

CICLO DE ESTUDOS
MESTRADO EM INFORMÁTICA MÉDICA

Validação da identificação de utentes num integrador de mensagens HL7 para monitorização e portabilidade de dados na área da saúde

Ana Cláudia de Loureiro e Nogueira

M

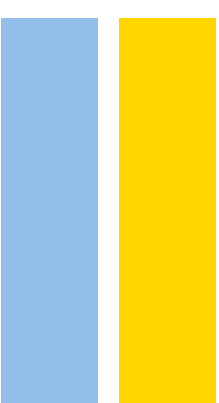
2019



SEDE ADMINISTRATIVA

FACULDADE DE MEDICINA

FACULDADE DE CIÊNCIAS





12ª ed

MIM

Validação da identificação de utentes num integrador de mensagens HL7 para monitorização e portabilidade de dados na área da saúde

Ana Cláudia de Loureiro e Nogueira

MESTRADO EM
INFORMÁTICA MÉDICA
2º CICLO DE ESTUDOS

SET | 2019

12ª ed

MIM

Validação da identificação de utentes num integrador de mensagens HL7 para monitorização e portabilidade de dados na área da saúde

Ana Cláudia de Loureiro e Nogueira

MESTRADO EM
INFORMÁTICA MÉDICA
2º CICLO DE ESTUDOS

ORIENTADORES

Pedro Manuel Vieira Marques
Ricardo João Cruz-Correia

SET | 2019

Agradecimentos

“A gratidão é o único tesouro dos humildes”

William Shakespeare

Ao Prof. Dr. Ricardo Cruz-Correia, pela partilha de conhecimento, a sua escuta ativa e crítica, pelas valiosas e sábias contribuições para o trabalho, assim como pelos esclarecimentos, orientação, correções e sugestões. Agradeço a confiança que depositou em mim e a abertura de caminhos na reflexão mobilizadora para o investimento neste trabalho.

Para o Doutor Pedro Manuel Vieira Marques pela sua disponibilidade que nos momentos certos soube dar ideias e perspetivas diferentes para a elaboração desta dissertação. Agradeço os longos cafés acompanhados de palavras-chave desencadeadoras de motivação, por me apontarem o caminho.

Aos elementos da empresa HLTSYS – HealthySystems, Lda.: Liliana Correia, Raphael Oliveira, Duarte Ferreira, Alexandre Santos, Ricardo Gonçalves, Alexandre Augusto, Rui Martins, Pedro Silva, Pedro Bastos, João Graça, Sandra Reis, Margarida Pinho e Isabel Cruz, pelas condições de trabalho proporcionadas e pela colaboração prestada que tornaram possível a concretização deste trabalho, proporcionando situações de aprendizagem do dia a dia.

Agradeço aos meus inestimáveis amigos Fátima Baldaia, Cristina Castro, Marina Guedes, Gabriela Felício, Joaquim Henriques, Rui Rafael e Pedro Vieira por acreditarem em mim, pelo vosso apoio e pelas vossas palavras de incentivo. Obrigada pelas horas e horas de conversas e dedicação que permitiram alcançar tudo o resto. Para todas as pessoas que colaboraram com um precioso contributo para o desfecho deste trabalho.

Aos meus pais que foram incansáveis e criaram condições para tornar este projeto possível. O meu emocionado muito obrigada!

Aos meus filhotes que são tudo para mim e por quem sou capaz de enfrentar o mundo para lhes proporcionar o melhor. A mamã esteve um pouco distante, mas espero ter a oportunidade de dar-vos o melhor de mim e o melhor do mundo.

“Põe o quanto tu és no pouco que fizeres.”
Fernando Pessoa

Resumo

No setor da saúde, o elevado número de Sistemas de Informação (SI) presentes nas instituições de saúde, a sua especificidade e a sua utilidade na melhoria da prestação de serviços de saúde, cria um problema sério quando existem erros nas integrações entre os mesmos. A qualidade dos dados é um aspecto crítico com alto impacto no processo de atendimento clínico e a qualidade da identificação do utente previne falhas na identificação e a falta de informação. Neste projeto pretende-se detalhar uma análise feita ao estado das integrações entre os SI de instituições hospitalares. Nestas instituições serão analisados os SI de alguns departamentos e as suas integrações com o sistema central do hospital. Estas integrações usam a norma *Health Level Seven* (HL7) para a comunicação de dados.

Background: A interoperabilidade entre diferentes SI em Saúde (SIS) tem sido um enorme desafio enfrentado pelas instituições da saúde nas últimas décadas, por precisarem de utilizar dentro de uma mesma instituição variados SIS e de diferentes fornecedores [1]. Apesar de se assistir a um esforço no desenvolvimento e adoção de *standards* com o intuito de facilitar o processo de interoperabilidade, continuam a existir diversas dificuldades na comunicação entre os sistemas, por vezes usando até os mesmos *standards* [2], o que pode afetar o funcionamento diário na prestação de cuidados hospitalares. O *Integrator for traceability in health* (HS.HELIOS) apresenta-se como uma solução de promoção da interoperabilidade entre os diferentes sistemas hospitalares, utilizando a tecnologia *opensource* (Mirth®) e o *standard* HL7, o mais utilizado em SIS [3], em linha com os perfis *Integrating the Healthcare Enterprise* (IHE) [1].

Métodos: Neste trabalho é usado um mecanismo de integração para receber, processar e reencaminhar mensagens HL7. Analisando essas mensagens, podemos fornecer os meios para superar a heterogeneidade presente nos modelos de dados e na arquitetura de sistemas de informação de saúde existentes. O objetivo deste trabalho foi o de criar uma solução de validação para identificação do utente usando como fonte mensagens HL7. A solução recebe um nome de utente e retorna informações sobre a sua qualidade.

Resultados: Num total de 1.048.576 mensagens foram recolhidas e processadas pela aplicação desenvolvida. Os testes realizados identificaram nomes de utentes indevidos (n=40.699) e também erros sistemáticos causados por alguns sistemas de informação em saúde. A solução também forneceu um método para aumentar a visibilidade desses problemas e permitir agir em conformidade para os corrigir.

Discussão: Num ambiente de produção, os testes realizados confirmaram a capacidade da solução para identificar erros comuns que acontecem entre as comunicações numa rede de instituições de saúde.

Os erros mais comuns detetados estavam relacionados com o campo do nome do utente que está a ser usado para outras funções além daquelas para as quais foi projetado.

Palavras Chave: *Standard HL7*, sistemas de informação de saúde, interoperabilidade, qualidade de dados, integração de dados.

Abstract

In the health sector, the high number of Information Systems present in health institutions, their specificity and their utility in improving health service delivery, creates a serious problem when there are errors in their integration. Data quality is a critical aspect with high impact on the clinical care process and the quality of patient identification prevents misidentification and lack of information. This project intends to detail an analysis made to the state of integrations between the IS of hospital institutions. In these institutions will be analysed the IS of some departments and their integration with the central system of the hospital. These integrations use the *Health Level Seven* (HL7) standard for data communication.

Background: Interoperability between different Information Systems in Health (HIS) has been a huge challenge faced by health institutions in recent decades, as they need to use multiple HIS and different providers within the same institution [1]. Although there is an effort to develop and adopt standards to facilitate the interoperability process, there are still several difficulties in communication between systems, sometimes even using the same standards [2], which may affect the daily functioning in the provision of hospital care. The Integrator for Traceability in Health (HS.HELIOS) is a solution for promoting interoperability between different hospital systems, using opensource technology (Mirth®) and the HL7 standard, the most commonly used in SIS [3], in line with the Integrating the Healthcare Enterprise (IHE) profiles [1].

Methods: In this work we used an integration engine to receive, process and route HL7 messages. By analysing these messages, we can provide the means to overcome the heterogeneity present in existing health information systems data models and architecture. The aim of this work was to create a validation solution for patient identification using HL7 messages as source. The solution accepts a patient name and returns information about their quality.

Results: A total of 1.048.576 messages were gathered and processed by the solution. The performed tests identified erroneous patient names (n=40.699) and also systematic errors

caused by some health information systems. It also provided a method to increase the visibility of these problems, and act accordingly to correct them.

Discussion: In a production environment, the tests performed confirmed the solution's ability to identify common errors that happen across communications in a health institution network.

Most common errors detected were related to the patient name field being used for other functions than those for which it was designed.

Keywords: Health Level Seven standard; health information systems; interoperability; data quality; data integration.

Resultados Científicos

Os resultados do trabalho descrito nesta dissertação foram publicados no seguinte artigo:

A. Nogueira, R. Oliveira, P. Vieira-Marques and R. Cruz-Correia, “Validation of patient identification in an HL7 messages integrator for health data monitoring and portability,” 2019 8th International Conference on Health and Social Care Information Systems and Technologies (HCist), Tunísia, 2019. À data da entrega desta dissertação, o artigo foi aceite e apresentado na *International HCist* que decorreu entre 16 e 18 de outubro de 2019.

Índice

Agradecimentos	vii
Resumo.....	ix
Abstract.....	xi
Índice.....	xv
Lista de Figuras.....	xvii
Lista de Tabelas.....	xvii
Acrónimos	xviii
1. Introdução	3
1.1. Estrutura da Dissertação.....	5
1.2. Objetivos.....	6
1.3. Pergunta de Investigação.....	7
1.4. Estado de Arte	8
1.4.1. Registo eletrónico de dados na saúde	8
1.4.2. Interoperabilidade na saúde.....	8
1.4.3. <i>Health Level Seven</i>	10
1.4.4. Mensagens HL7 em ambiente hospitalar.....	13
1.4.5. Algoritmos de apoio à decisão	15
1.4.6. Portabilidade de dados digitais.....	16
1.4.7. Qualidade de dados dos utentes.....	17
1.4.8. Trabalhos Relacionados.....	18
2. Sistema de Validação de Nomes de Utentes.....	23
2.1. Desenho do Estudo.....	23
2.2. Instrumento de recolha de dados	26
2.3. Metodologia de análise dos dados	26
3. Métodos	28
3.1. Ética.....	29

3.2.	Recolha e transformação de dados.....	30
3.3.	Descrição do algoritmo	31
3.4.	Avaliação da qualidade dos nomes dos utentes nas mensagens.....	34
4.	Resultados	39
5.	Discussão.....	47
6.	Conclusões e Trabalho Futuro.....	51
6.1.	Conclusões	51
6.2.	Trabalho Futuro	51
7.	Referências.....	55
8.	Anexos.....	63
8.1.	Algoritmo Name Quality na Linguagem Javascript.....	63
8.2.	Algoritmo na Linguagem Java de Validação de Nomes de Utentes	67
8.2.1.	NameQuality.java	67
8.2.2.	CheckName.java.....	70
8.2.3.	BaseFiles.java	81
8.2.4.	Count.java	89
8.2.5.	Output.java.....	94

Lista de Figuras

Figura 1.1: Esquematização da estrutura da investigação.....	5
Figura 1.2: Importância da integração para as organizações versus complexidade [31].	9
Figura 1.3: Arquitetura geral de comunicação HL7 com um sistema multiagente [45].	11
Figura 1.4: Representação de uma solução em que vários SI comunicam com base no Global Master Patient Index [50].	13
Figura 3.1: Arquitetura geral do sistema de <i>HS.HELIOS System</i>	31
Figura 4.1: Apresentação do <i>dashboard</i> criado para visualização da quantidade de nomes detetados com erros.	42
Figura 4.2: Apresentação do <i>dashboard</i> que visualiza a quantidade de nomes detetados com erros, agregados por nível de qualidade.	42
Figura 4.3: Esquema do algoritmo.	44

Lista de Tabelas

Tabela 1: Sistemas de informação de saúde instalados nas duas instituições.	23
Tabela 2: Frequência e percentagem para a análise LNL e LNI num total de 1 048 576 mensagens.	40
Tabela 3: Frequência e taxa de erro nas duas iterações num total de 1.048.576 mensagens.	41

Acrónimos

ADT	Admission Discharge-Transfer
ANSI	American National Standards Institute
API	Application Programming Interface
CORBA	Object Management Group
DCOM	Distributed Component Object Model
DICOM	Digital Imaging and Communications in Medicine
EHR	Electronic Health Record
ESB	Enterprise Service BUS
GMPI	Global Master Patient Index
HELIOS	Integrator for traceability in health
HI	Health Institution
HIE	Health Information Exchange
HIS	Health Information Systems
HL7	Health Level Seven
IHE	Integrating the Healthcare Enterprise
IRA	Integrated Routing Audit
IT	Information Technology
JSON	JavaScript Object Notation
LNI	Lista de Nomes Inválidos
LNL	Lista de Nomes Legais
MLLP	Minimum Lower Layer Protocol
openEHR	open standard specification in health informatics for EHRs
OSI	Open Systems Interconnection
PID	Patient IDentification Segment
PIB	Produto Interno Bruto
RGPD	Regulamento Geral sobre a Proteção de Dados
SI	Sistema de Informação
HDN	Health Data Network (Rede de Dados de Saúde)
SNS	Serviço Nacional de Saúde
XPN	eXtended Person Name



Introdução

1. Introdução

O setor da saúde tem registado uma evolução notável ao longo das últimas décadas, sendo esta uma área em constante desenvolvimento em que a gestão da informação assume um papel fulcral nos Sistemas de Informação (SI) instalados nas Instituições de Saúde (HI). Entende-se por SI um sistema automatizado, ou manual, que deve ser capaz de recolher, processar, transmitir e difundir dados. Nas instituições de saúde coexistem vários SI, com o intuito de facilitar as integrações do volume crescente de informação, para tal estes sistemas utilizam tecnologias e normas diferentes. O desenvolvimento dos SI pretende promover a eficiência das organizações e é necessário que todos os intervenientes neste processo caminhem na mesma direção, melhorando a documentação, comunicação e coordenação entre sistemas.

O domínio da saúde tornou-se um setor altamente dinâmico onde muitas informações sobre os utentes são armazenadas e utilizadas durante os diversos fluxos de trabalho dos profissionais de saúde. Ao longo dos anos, a qualidade dos dados nos sistemas de informação em saúde tornou-se cada vez mais evidente. Foram desenvolvidos novos modelos de dados que adotam políticas baseadas nas necessidades dos utentes, a fim de promover a qualidade e a segurança nos cuidados prestados aos utentes [2]. Por outro lado, existe uma carência de investigações relacionadas à introdução dos registos eletrónicos de saúde (EHR - *Electronic Health Records*) e ao seu subsequente impacto na qualidade dos dados e na prestação de cuidados. Adler-Milstein et al. afirmaram que a "qualidade dos dados" na área de cuidados de saúde apresenta problemas, identificando a ausência de uma definição padronizada [3].

A interoperabilidade entre o domínio da saúde é de enorme necessidade, mas, ao mesmo tempo, é muito complexa, onde um dos seus maiores desafios é encontrar uma solução para partilhar informações entre diferentes sistemas sem perda de sentido ou mesmo propagação do erro [1], [4]–[6] [5]. Essa complexidade pode levar a comunicações inconsistentes e à falta de confiança dos utilizadores no resultado dos sistemas. Há muitos benefícios na interoperabilidade [4]–[6], incluindo:



Para que seja alcançada a interoperabilidade, necessita-se de garantir que a identificação do utente seja de qualidade e se torne cada vez mais pertinente para reconhecer corretamente e sem dúvida o utente entre os diversos SI em saúde. Os problemas proprietários de fornecedores são uma barreira à interoperabilidade e à inovação. A atual falta de interoperabilidade entre os recursos de dados para o EHR, torna-se um grande impedimento para a troca de informações sobre saúde e o desenvolvimento de uma infraestrutura robusta de dados. Os problemas de interoperabilidade para informações de saúde podem ser resolvidos apenas estabelecendo uma arquitetura de *software* abrangente, transparente e global [7].

Devido à individualidade dos diferentes sistemas e configurações correspondentes, a identificação do utente pode ser introduzida em vários momentos e/ou em vários sistemas. Normalmente não há uma uniformização ou um único processo de inserção de dados de identificação do utente. Como consequência, surgiu a necessidade da realização de um estudo sobre a qualidade da identificação do utente, em particular o nome do utente. Esta

identificação baseia-se na implementação de uma técnica em contexto real de uso em instituições de saúde. Assim, centralizam-se as transmissões de dados clínicos e demográficos, permitindo a extração de métricas relacionadas a integrações. Permite ainda, a verificação da integridade semântica das mensagens HL7 trocadas, com a respetiva monitorização em tempo real das integrações.

1.1. Estrutura da Dissertação

De forma esquemática a dissertação está organizada da forma como se representa na Figura 1.1. O estado de arte é constituído pela componente crítica das bibliografias que fundamentam o trabalho, tópicos reagrupados segundo temas tratados: registo eletrónico de dados, interoperabilidade, HL7, mensagens HL7, algoritmos de apoio à decisão e qualidade dos registos dos utentes. Nesta secção examinam-se também os instrumentos de medida e as técnicas de análise de cada investigação desejada compreender, correspondente ao processo de evolução tecnológica e das publicações científicas. Descrevem-se os elementos comuns e os divergentes para concretização das etapas da conceptualização da investigação, para inovação através de uma aplicação tecnológica programada para o efeito.

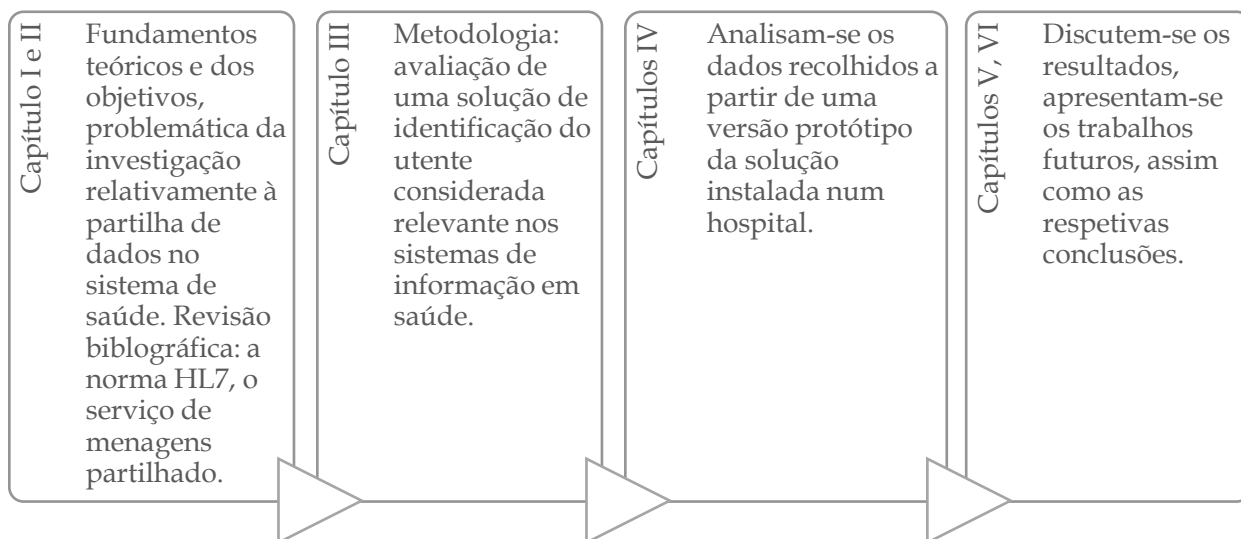


Figura 1.1: Esquematização da estrutura da investigação.

Em relação ao problema, sustenta-se o documentado pelos diversos investigadores relativamente ao conflito no sistema de mensagens HL7 na saúde. O estudo bibliográfico dá ênfase à seleção dos conceitos que compõem este estudo ou as teorias que levam a conclusões

provadas sobre as melhorias dos efeitos de uma nova aplicação de algoritmos para a interoperabilidade digital dos sistemas de mensagens em hospitais.

1.2. Objetivos

Este projeto tem por objetivo conhecer a situação atual ao nível das integrações entre SI existentes, nomeadamente análise de *workflow* institucional e análise de mensagens, e verificar os vários estados das mensagens trocadas num dado *workflow*, bem como a própria estrutura e sintaxe dos dados transmitidos, através da análise aos segmentos (e campos) que compõem as mensagens HL7.

Num primeiro momento desenvolve-se um sistema que tire partido das mensagens trocadas entre diferentes SI. O sistema sustenta: a) a monitorização: monitorizar em tempo real a produção de informação pelos vários departamentos da instituição; b) a qualidade de dados: identificar falhas na identificação dos utentes; c) a redundância: criar fontes de dados e alimentar arquivos digitais centrais; d) o cruzamento de dados: relacionar dados provenientes de diferentes sistemas. Pretende-se ainda que o sistema consiga responder aos seguintes requisitos: 1) tolerância a falhas: a construção de registos com informação demográfica; 2) normalização de comunicação: a implementação de mecanismos de obtenção de dados utilizando padrões *Integrating the Healthcare Enterprise* (IHE); 3) independência: a criação de bases de dados passíveis de ser controladas pelo próprio hospital.

O projeto não teve qualquer impacto na performance dos sistemas e comunicações já existentes. O sistema assenta na capacidade de analisar diretamente a rede de comunicações das mensagens trocadas, não havendo necessidade de proceder a alterações nos sistemas de informação emissor e recetor envolvidos na troca de mensagens.

A área da saúde tornou-se um setor em constante evolução, especialmente no que diz respeito aos SI que suportam o processo de assistência à saúde [5], [8]–[14]. Neste setor, a qualidade dos dados é um aspeto crítico, com alto impacto nos processos de cuidados, particularmente quando envolve a identificação de utentes. Em muitas instituições, os dados estão espalhados por vários sistemas e estão a aumentar a dificuldade de monitorizar a qualidade dos dados. Com este projeto pretende-se centralizar o controlo a falhas.

Este trabalho tem como objetivo a criação de uma componente de validação para identificação de utentes em mensagens HL7 que são trocadas entre SI em saúde de instituições hospitalares. A abordagem apresentada aceita o nome do utente e retorna

informações sobre a qualidade do nome a ser processado no mecanismo de integração e permite realizar um estudo estatístico dos erros mais comuns encontrados.

Neste trabalho, usamos o mecanismo de *interface* multiplataforma *Mirth Connect*® como um roteador de gestão para processar automaticamente as mensagens HL7 - Versão 2.x. Monitorizando essas mensagens trocadas, podemos fornecer os meios para superar a heterogeneidade nos sistemas de saúde existentes. De um conjunto de dados de 1.048.576 nomes, que estiveram em produção durante 12 dias, identificaram-se os nomes que estavam errados. O algoritmo revelou que, com alguns nomes inválidos sendo adicionados à Lista de Nomes Inválidos (LNI), um grande número de nomes incorretos foi detetado. Concluimos que tem uma grande sensibilidade, pois os nomes corretos são mais detetados do que os que não são.

1.3. Pergunta de Investigação

As questões de investigação precisam as variáveis utilizadas na mesma, assim como as relações que possam existir entre elas, especificando os aspetos a estudar [15]. Portanto, a pergunta de investigação foi elaborada com base na análise da metodologia PICO, que se descreve em seguida o seu significado:

- **População:** Centros hospitalares.
- **Intervenção:** Integração de mensagens HL7 em SI de saúde.
- **Controlo:** Sem integração e sem monitorização.
- **Outcome:** Monitorização e portabilidade de dados de saúde.

Dessa forma, os princípios sobre as quais se apoia o resultado, abarcam a seguinte pergunta de investigação:

“Um integrador de mensagens HL7 para a monitorização e portabilidade de dados de saúde entre centros hospitalares aumenta a eficiência dos SI de saúde em comparação com o método tradicional?”

A questão de investigação tem a ver com os principais conceitos estudados, como sejam: HL7, saúde, mensagens, interoperabilidade, portabilidade, utente e nome. Com esta questão de investigação tratam-se as relações entre as variáveis e testa-se um modelo de portabilidade dos dados de saúde. Especificamente testa-se o sistema de partilha de mensagens entre utilizadores autorizados. No modelo digital criado consta: a aplicação criada através de

algoritmo, como suporte à interoperabilidade e a ser adotado nos sistemas da saúde. Portanto, os erros são explicados devido à partilha das mensagens HL7 e à utilização indevida do campo de identificação do utente.

1.4. Estado de Arte

1.4.1. Registo eletrónico de dados na saúde

Nos Estados Unidos o uso generalizado EHRs de dados sobre a saúde dos cidadãos considera-se inevitável [16]. Os benefícios da tecnologia foram identificados quer pelos utentes quer pelos médicos durante as práticas profissionais, sendo ainda confirmado por diversos investigadores [12], [13]. A investigação tem demonstrado concordância profissional e consenso da partilha em relação ao valor potencial dos EHRs [16], [17]. Os investigadores invocam a disponibilidade de EHRs de algumas grandes instituições, desde os hospitais, às clínicas privadas ou mesmo a partilha com os fornecedores [18].

A mudança para a partilha eletrónica de informações de saúde tem como consequência a melhoria dos cuidados de saúde a custos mais reduzidos [9]. A assistência à saúde é um dos maiores segmentos da economia dos EUA, com um índice aproximado de 20% do Produto Interno Bruto (PIB) [19]. Os aspetos tecnológicos digitais para a medicina moderna, permitem que cada utilizador aceite a partilha e indique quem tem acesso aos seus dados clínicos [20]–[22]. A aposta nos registos eletrónicos em medicina tem sido opção por parte dos administradores de serviços de saúde e pelos governos, nomeadamente os ministérios com responsabilidade pela assistência à saúde [23]. A transição para os registos eletrónicos apresenta muitos desafios. Esta transição precisa encontrar um equilíbrio entre a proteção da privacidade do utente e os benefícios que o uso partilhado de dados oferece para melhorar os resultados na saúde e reduzir custos [23].

1.4.2. Interoperabilidade na saúde

A gestão de dados digitais na área da saúde deve ser clara para todos os envolvidos, permitindo com isso evitar erros. A interoperabilidade dos dados é um pilar no apoio à medicina digital emergente, a saúde eletrónica e a medicina preditiva, preventiva, personalizada e participativa, chamada de P4 [6]. O sistema, com diversas aplicações, requer garantia na partilha de informações de utentes, oportunas, precisas e seguras [24]. A

fragmentação das aplicações traz dificuldades na transmissão das informações entre os serviços [24], [25]. Diversos estudos têm informado que os utilizadores das infraestruturas digitais têm referido dificuldades em obter informações precisas [24], [26]–[28]. Nesse sentido, a qualidade dos dados torna-se uma preocupação [29].

A partilha e a centralização dos cuidados entre hospitais é cada vez mais comum em sistemas de saúde em todo o mundo [24]. Essa partilha dos EHRs tornou-se universal [30].

A integração dos dados ligada à interoperabilidade pode ser classificada em quatro perspetivas tecnológicas mais comuns e abrangentes[31], discriminadas:

- Integração da Informação (II): centraliza-se na informação, a sua gestão e disponibilidade;
- Integração Aplicacional (IA): nesta perspetiva a integração das aplicações são o objetivo primordial;
- Integração de Processos (IP): a integração de SI feita numa lógica processual;
- Integração inter-Organizacional (IO): a informação e o formato da sua troca entre organizações são o objetivo desta perspetiva.

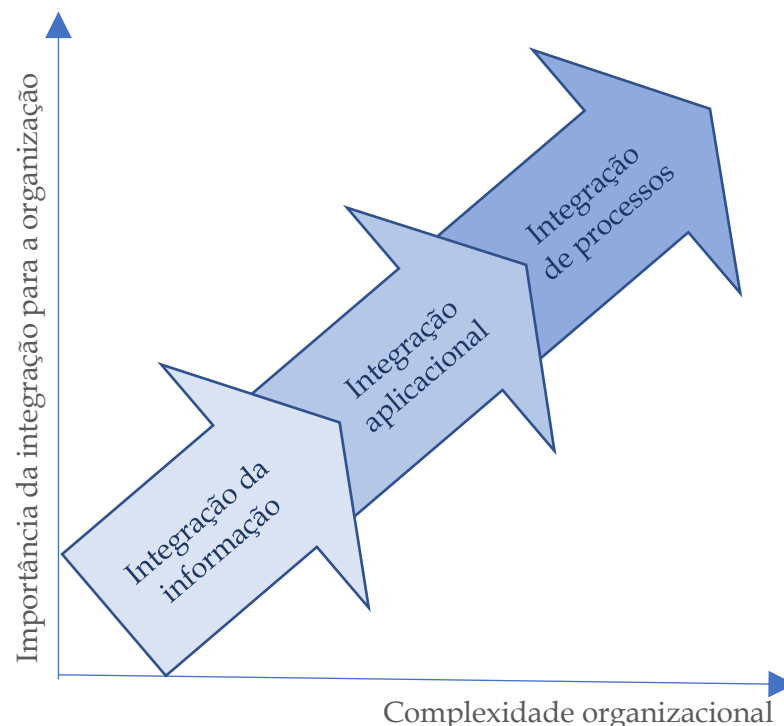


Figura 1.2: Importância da integração para as organizações versus complexidade [31].

No entanto, a interoperabilidade do EHR continua a ser um desafio, particularmente entre os fornecedores, e a partilha de informações torna-se rara, apesar do investimento do governo [32]. A privacidade pode ser uma preocupação crítica durante as decisões de adoção dos EHR [33]. Para evitar negligenciar os EHR deve-se garantir que os clínicos possuam

conhecimento de todo o processo [34]. Além destas situações, a questão que se coloca prende-se com a baixa interoperabilidade dos sistemas de registo relacionados à saúde entre as organizações [35]. Como diversos investigadores detetaram que a interoperabilidade nos hospitais precisa de ser melhorada, serve de prova de que existe a necessidade de fornecer uma capacidade alternativa e eficaz de partilha de dados para a qualidade dos cuidados aos utentes [24], [35]. Os investigadores [24], [30] identificaram benefícios para uma melhor interoperabilidade em que a qualidade do atendimento aumenta. Além disso os erros médicos são um problema generalizado com sérias repercussões na morbilidade e mortalidade dos utentes [36]. Mais provas que permitem afirmar a necessidade de investir na interoperabilidade. Portanto, já foi identificada a necessidade de maior intercâmbio de informações sobre saúde e como os EHRs podem servir como plataformas para a inovação baseada em dados e podem ser ideais para a aplicação de métodos de *Big Data* [30], [37]. *Big Data* significa grandes volumes de dados e com grande diversidade da tipologia de dados [29]. Os investigadores estimaram, relativamente aos dados clínicos, atingir 80 a 100 PetaBytes até 2025 [29], [30].

1.4.3. *Health Level Seven*

Os processos de transferência de informação digital podem estar padronizados ou standardizados. Um padrão pode ser definido como um conjunto de regras e definições que especificam como transferir um processo ou produzir um produto [38]–[40]. Um padrão é útil para fornecer uma solução para um problema e este pode ser utilizado sem ser preciso começar de raiz, uma vez que oferece bases de conhecimento que foram previamente estudadas. A utilização dos padrões facilita o trabalho de forma colaborativa [38]–[41]. Alguns padrões evoluem ao longo do tempo ou são desenvolvidos deliberadamente [41]. As normas são geralmente necessárias quando a diversidade excessiva cria ineficiências ou impedem a eficácia.

As integrações entre os SI usam a norma HL7 [42]. A HL7 abarca um conjunto de normas internacionais para a representação e a transferência de dados clínicos e administrativos. Para a sua concretização, o registo digital da informação torna-se indispensável. HL7, além do *standard*, dá o nome a uma organização internacional certificada pela *American National Standards Institute* (ANSI). Esta organização foi fundada em 1987, sem fins lucrativos, e desenvolve normas, especificações, protocolos ou padrões que devem ser utilizados para a área de saúde na troca de dados entre SI, desde 1994 [43]. Especificamente, no *standard* HL7, tratam-se a trocas de mensagens entre SI. A referência ao nível 7 significa o mais alto nível de aplicação presente no modelo de comunicações para interconexão entre sistemas *Open*

System Interconnection (OSI) [44]. Os serviços da aplicação definem as mensagens e os mecanismos de troca de dados.

Uma vez que esta troca de informação é de natureza hospitalar, ou seja, inerentes à saúde, pode-se relacionar a norma HL7 com o conceito de *Health Information Exchange* (HIE) - Intercâmbio de Informação Hospitalar. Esta relação é possível dado que, este conceito se baseia na interoperabilidade, do mesmo modo que a norma HL7 foi implementada para alcançar a interoperabilidade. Por outras palavras, como forma de assegurar a interoperabilidade entre os Sistemas de Informação Hospitalares (HIS) implementa-se a norma HL7, que permitem a interpretação da informação por todos os sistemas envolvidos [5]. Portanto, a técnica HL7 torna-se crucial para alcançar a interoperabilidade nos cuidados de saúde [5], [8].

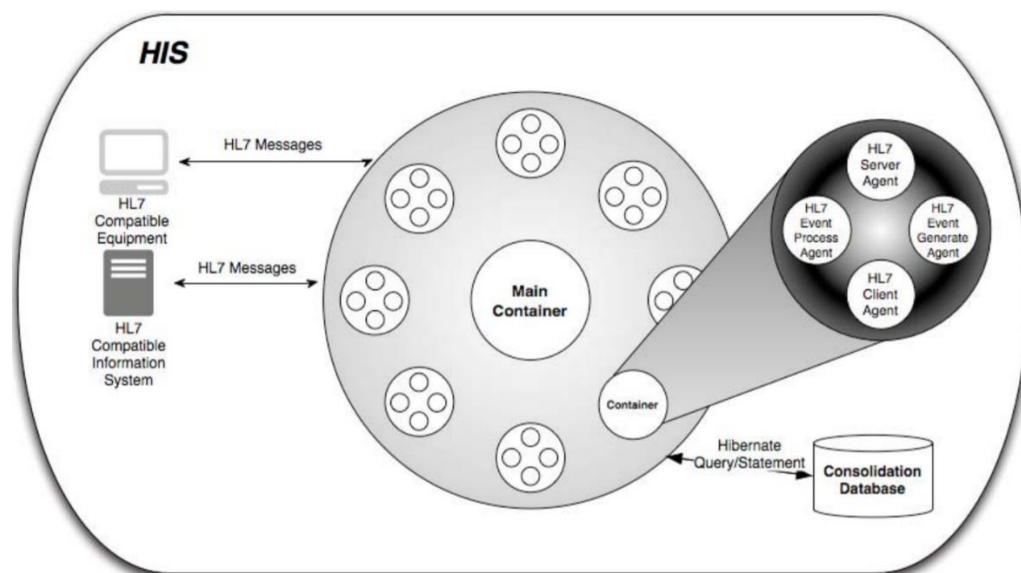


Figura 1.3: Arquitetura geral de comunicação HL7 com um sistema multiagente [45].

A Figura 1.3 ilustra as fases de uma integração geral de SI de comunicação utilizando a norma HL7 com um sistema multiagente. As variáveis estão relacionadas a essas etapas e pretendem descrever o contexto em que ocorre a integração (recetor principal, equipamento compatível HL7, SI compatível HL7, mensagens HL7, recetor externo HL7: eventos, processos e agentes; servidor e agente; eventos, criar um novo registo, agente; cliente e agente), o tipo de sistema de comunicação para consolidação da base de dados.

A estruturação e *design* do padrão HL7 (Figura 1.3), definindo quais os artefactos de dados devem ser transferidos por uma certa mensagem, permitindo e potenciando a aplicação do HL7 numa arquitetura cliente-servidor [8], [46]. A versão HL7 v2.x tem algumas limitações:

a demora da sua implementação, também pelo facto de não existir um processo rigoroso para a sua execução e também existirem demasiadas opções e situações ambíguas [47]. O HL7 v2.x ainda hoje é o padrão mais amplamente disseminado e implementado [48].

As principais missões do protocolo HL7 referem-se ao desenvolvimento de sistemas e normas vocacionadas para a troca, integração, partilha e recuperação de informação eletrónica na saúde, para suporte da prática médica e administrativa, possibilitando um maior controlo dos serviços de saúde, e por outro lado, criar metodologias, padrões e diretrizes que sejam flexíveis, viáveis economicamente e que permitam a interoperabilidade e partilha de informações clínicas digitais [8], [46], [46], [47]. Pode-se dizer que a norma HL7 fornece diretrizes para a troca de dados entre SI em saúde, ajudando a reduzir os erros de comunicação e os problemas de qualidade dos dados, aumentando assim a interoperabilidade numa instituição de saúde, ajudando a reduzir os erros de comunicação e os problemas de qualidade dos dados, aumentando assim a interoperabilidade numa instituição de saúde.

Tradicionalmente, o ambiente na área da saúde tem consistido num conjunto de unidades livremente conectadas e independentes dentro de cada organização. Os utentes recebem atendimento nos diversos serviços de saúde primária, secundária e terciária, com pouca comunicação bidirecional e falhas na coordenação entre esses serviços. Quando Ferreira et al. afirmam que [49, p. 197] *"além da desvantagem óbvia e custos adicionais associados com a necessidade de armazenar várias informações em diferentes locais, tal fragmentação também significa que as instituições de saúde, muitas vezes, têm as mesmas informações armazenadas com diferentes formatos e em diferentes sistemas, o que implica a complexidade extra de manter todas essas versões atualizadas e consistentes entre si"*. Denota-se que partilham a preocupação com a falta de uniformização dos dados. Esta situação coloca em causa a interoperabilidade numa instituição de saúde, podendo levar a desafios complexos de integração que não podem ser facilmente resolvidos sem a intervenção dos fornecedores de *software*. Detetadas as situações de conflito tecnológico, os utilizadores expressam as dificuldades com os fornecedores do *software* para possíveis correções de integração nos sistemas de produção com um maior conhecimento sobre a realidade nas integrações [49].

Sistemas digitais com computadores separados podem participar de uma Rede de Dados de Saúde (HDN), conectados para partilhar vários tipos de informações relacionadas à assistência médica (Figura 1.4). As informações partilhadas podem incluir informações de registo do utente.

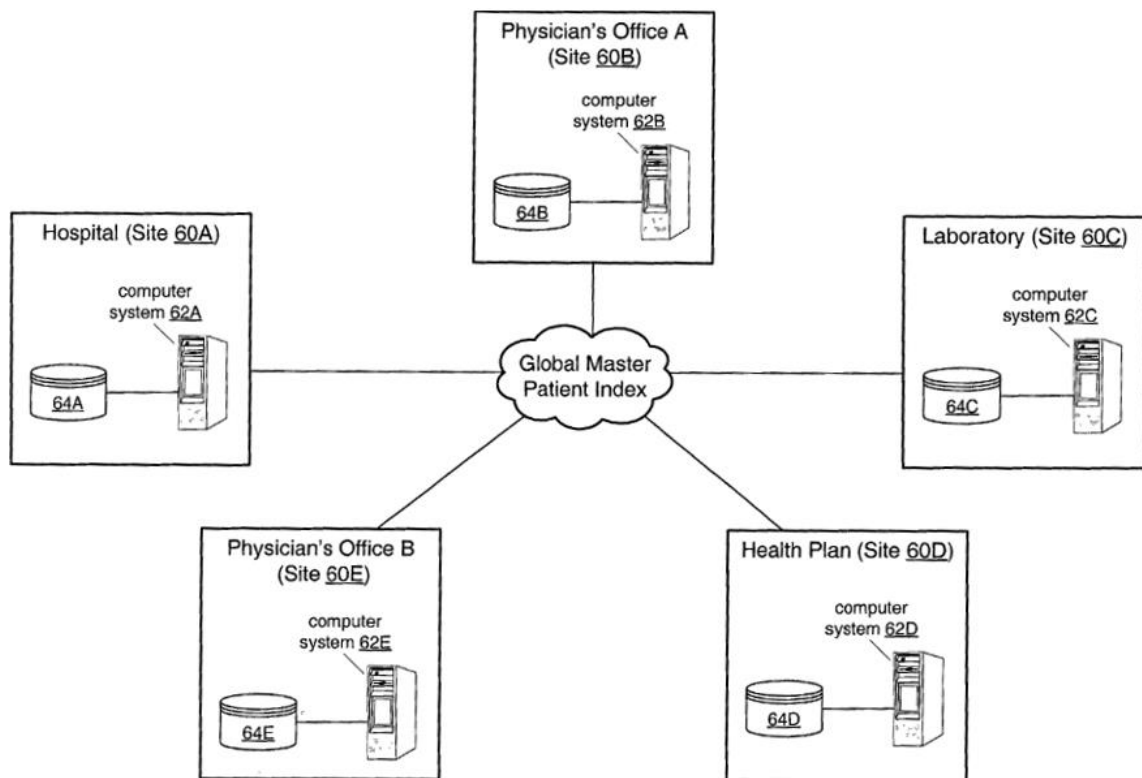


Figura 1.4: Representação de uma solução em que vários SI comunicam com base no Global Master Patient Index [50].

A integração das informações do registo do utente realiza-se através da manutenção de um *Global Master Patient Index* (GMPI) [50]. Esse GMPI pode integrar informações de registos de utentes. Estes dados podem ser utilizados por várias organizações públicas e privadas, como as empresas de saúde [51], [52]. Esse GMPI também pode integrar informações de registo do utente para uma única empresa que possui vários sites ou sistemas de computadores, por exemplo, um grande hospital. Através das funções do GMPI consegue-se as indicações de como localizar registos de utentes, localizar registos duplicados, imprimir, corrigir possíveis registos duplicados e guardar as atualizações dos dados (certificação) [53]. Assim, mantém-se um relacionamento persistente entre os registos de utentes no GMPI, procedendo a auditorias para possíveis correções na identificação de cada cidadão registado como utente [50].

1.4.4. Mensagens HL7 em ambiente hospitalar

Os serviços de saúde em Portugal têm adotado a norma HL7 como protocolo de comunicação entre sistemas, para troca de informações, aumentando a interoperabilidade entre os SI das instituições de saúde [5], [49]. A interoperabilidade permite que seja possível diminuir a fragmentação, aumentar a qualidade dos dados clínicos, tornando os sistemas

mais homogêneos [54]. Para trocas de mensagens entre SI, a norma HL7, está presente na última camada do modelo *Open System Interconnection* (OSI) [44], onde os serviços da aplicação definem as mensagens e os mecanismos de troca de dados. A interoperabilidade vê-se como um fator crucial para a viabilidade do modelo Health@Home [55]. O modelo Health@Home integra a prevenção de doenças e cuidados de saúde para diferentes grupos de cidadãos em diferentes fases da escala de níveis de prestação de cuidados na saúde [1], [26]. O lema deste modelo abarca a facilidade de utilização para os cidadãos [27].

Uma mensagem HL7 v2.x contém informações demográficas que são usadas para ajudar a identificar um determinado utente, tais como: nome, endereço, número de segurança social e data de nascimento. Essa abordagem é feita para facilitar a flexibilidade e a interoperabilidade em implementações nos vários SI em saúde entre um cenário de uma instituição ou entre instituições [48]. No caso de mensagens HL7 v2.x, as informações de um utente são descritas no segmento *Patient IDentification Segment* (PID), mais concretamente o nome do utente está presente no campo PID-5 e o respetivo tipo de dados é *eXtended Person Name* (XPN). Este campo contém vários componentes e a solução descrita no documento usa o seguinte: (1) nome de família, *Family Name* (FN), (2) nome próprio, *Given Name* (GN) e (3) segundo e nome posterior, *Second and Further Given Name or Initial Thereof* (ST) [56]. Os dois primeiros campos são necessários para a solução desenvolvida e descrita neste trabalho.

Botsis et al. em [57] analisaram EHRs do centro médico da Universidade de Columbia e afirmam que a falta de dados completos pode muitas vezes acontecer principalmente por duas razões: (a) fragmentação de dados causada por tratamentos de utentes que são feitos em diferentes instalações de cuidados de saúde, e (b) falta de informações contextuais causadas por má documentação médica. Logo, as mensagens seguem incompletas. Como exemplos de tais problemas foram detetados em segmentos HL7 PV1, onde as datas estavam a ser agregadas com o número de identificação de visita ou mensagens *Admission Discharge-Transfer* (ADT) contendo códigos numéricos para definir o local de nascimento de um determinado utente que, de outra forma, deve ser representado como uma sequência de caracteres [49], [58].

Com o propósito de criar um mecanismo capaz de monitorizar as questões apresentadas, pode-se tirar proveito do sistema *Integrated Routing Audit* (IRA) para HL7 [49]. Os investigadores referem que, com o IRA coleta-se diretamente os dados da rede, gere-se centralmente e avalia-se em tempo real a qualidade dos dados dentro das mensagens HL7 trocadas. Portanto, pode-se estender as funcionalidades atuais do IRA HL7, criando uma ferramenta automatizada capaz de analisar em tempo real a qualidade dos dados presentes nas mensagens HL7. Além de todas as necessidades identificadas, padronizar a terminologia

das mensagens vê-se como a resolução de uma lacuna no sistema, a semântica e a sintaxe, podendo ser solucionado com um padrão codificado [59].

O *Clinical Document Online Translator* (CDOT) tem como objetivo fornecer os meios para transformar dados provenientes de mensagens HL7 v2.x em composições *openEHR Foundation* (openEHR) [41]. Assim, pretende-se que essas composições possam ser usadas para alimentar um arquivo único e bem estruturado de dados clínicos. Para implementar esse sistema, utilizamos uma instalação do mecanismo de integração do *Mirth Connect* [10] e estruturas de dados *JavaScript Object Notation* (JSON) para comunicação [4]. Através do roteador de mensagens (*message broker*), a solução CDOT recebe como entrada mensagens HL7 v2.x e analisa cada uma delas, mapeando os dados clínicos relevantes para uma variável JSON [4], [41]. Agindo como um intermediário, a variável JSON utiliza-se para preencher as composições do openEHR [41]. Alguns problemas têm sido resolvidos com um módulo de *Sniffer* HL7. Este foi desenvolvido com uma estrutura de código aberto para *sniffers* de tráfego na linguagem GO (criada pelo google) [60], permitindo analisar e extrair o conteúdo da mensagem HL7 no protocolo *Minimum Lower Layer Protocol* (MLLP) [49], [60]. A abordagem, permite inspecionar com segurança todo o tráfego HL7 em todos os SI que contenha dados relevantes.

As tecnologias por mensagens, como as HL7, são mais amplamente utilizadas do que as soluções de *middleware*, que é *software* com múltiplas funcionalidades que fornece serviços para aplicações (como o *Distributed Component Object Model* [DCOM] ou o *Object Management Group* [CORBA]) [61].

1.4.5. Algoritmos de apoio à decisão

Os algoritmos ou sequência de instruções de apoio à decisão podem identificar e evitar erros, como os que se dão na área da saúde [36]. Os sistemas de suporte à decisão mais comuns são projetados para evitar erros, como por exemplo de medicação [30]. Podem ajudar a implementar diretrizes de cuidados e minimizar o desvio a práticas desapropriadas ao utente [62]. O EHR pode resolver os erros de comunicação, como da legibilidade da caligrafia [30].

Os EHRs processados em tempo real já permitem dar um alerta de suporte a decisões clínicas baseado na melhor adesão às diretrizes de tratamento e os desfechos clínicos em utentes hospitalizados [63]. Referem ainda que, existem uma variedade de plataformas de *software*: EHRs desenvolvidos internamente nos hospitais, com programas de *software* autónomos e EHRs comerciais [63]. No entanto, diversas investigações referem que nem

todas essas plataformas tiveram impacto no processo de adesão ou mesmo como medida a implementar ou implementada [64]–[66]. A qualidade de dados e complexidade técnica evidenciam desafios [67]. Estas descobertas inconsistentes podem refletir fragilidades metodológicas na bibliografia existente, baseadas em poucos estudos, sem os validar interna e externamente, ou refletindo as diferenças na implementação, como algoritmos de alerta e cuidados associados [63]. A maioria dos centros médicos não pode investir na construção dos modelos digitais de registo de dados, comprar ou recorrer ao suporte de sistemas de alerta independentes [63]. Muitas infraestrutura de tecnologia da informação apresentam um *design* de alerta limitado às funcionalidades padrão de EHR e dar o alerta é fundamental para um impacto clínico positivo [63], [68].

Os algoritmos também já foram criados para aumentar a capacidade dos investigadores para usarem o *big data* em utentes oculares, a investigação de Stein et al. [69], em oftalmologia, durante 1 de agosto de 2012 e 31 de agosto de 2017. Os investigadores referiram que a probabilidade de cada utente ter síndrome de esfoliação foi estimada usando a regressão logística de menor encolhimento absoluto e operador de seleção. A utilização de dados para detetar algoritmicamente eventos relacionados à assistência médica em tempo real tornou-se uma realidade [67]. As aplicações com algoritmos usam-se em dados de saúde, além de fatores já apontados neste estudo, para detetar eventos, particularmente tentar identificar preocupações e erros de segurança em tempo útil [67].

1.4.6. Portabilidade de dados digitais

A portabilidade de dados digitais permite que estes sejam processados e trocados no momento. Os aplicativos que oferecem portabilidade, com um painel para mostrar os dados sobre o tipo de tratamento de utentes são uma realidade no mercado [73]. Desde o aparecimento dos *smartphones* que a barreira da acessibilidade ficou reduzida [14]. Um dispositivo móvel é um sistema flexível e ágil por natureza, que permite responder rapidamente às necessidades [74], através da receção de dados por *Wi-Fi* [75]. Estes métodos permitem realizar a leitura e a correlação de sinais com processamento de dados em tempo real [75]. O sistema de registo de dados, pode ainda ser efetuado por um processo de reconhecimento de voz. A *Mobile Health*, ou '*mHealth*', é a aplicação de dispositivos móveis, com componentes e tecnologias relacionadas aos serviços de saúde [14]. Os investigadores referidos [14] vão mais longe, quando defendem que a utilização de aplicativos móveis permitem que a capacidade das autoridades de saúde pública façam uma gestão mais controlada na eventualidade de surtos. Muitas das informações ou dados podem ser disponibilizados em nuvem por forma a facilitar o acesso fora do hospital. Os dados, na sua

maioria, são disponibilizados aos utilizadores através de um navegador da *Web*, podendo ser utilizado em diversos aplicativos. Os acessos *Web* apresenta-se com 92%, e aplicativos cliente-servidor com 19% [61]. Os dados disponibilizados para outro SI recorrendo ao servidor *Web* já contam com 88%, os que utilizam o CORBA, 13% [61]. A implementação de um quadro com uma estrutura semântica no EHR, facilita o rápido desenvolvimento e a portabilidade de soluções de TI em saúde [59].

1.4.7. Qualidade de dados dos utentes

Um grande desafio nos sistemas de informação em saúde é poder reunir as diferentes partes do registo de um utente, sem qualquer risco de trocarem a identificação dos utentes [76]. É essencial fornecer cuidados consistentes aos indivíduos - centrados no utente [61]. Muitas vezes, os utentes são registados incorretamente. A qualidade dos dados dos utentes tem sido bastante baixa em vários sistemas de informação em saúde. Esse problema é cada vez mais evidente à medida que se registam mais dados sobre os utente [77].

No estudo de revisão sistemática de Correia-Cruz et al. (2007) [72], em 56 projetos distintos, sendo a sua maioria à escala regional, a integração foi mais comumente realizada por meio de mensagens com modelos predefinidos e soluções de *middleware*. De acordo com os referidos investigadores, identificaram que os dados médicos compartilhados nos projetos foram: diagnóstico e problemas, com 77%, imagens médicas, com 67% e resultados laboratoriais, com 65%. As soluções tecnológicas distintas, usando diferentes padrões e arquiteturas de dados podem dificultar a interoperabilidade [61], [78]. Sendo que, tendo como referência o período entre 2003-05, a maioria das informações integradas vem do hospital (69%), por departamento (40%) e cuidados primários (33%) [61].

Por vezes, os erros de identificação do utente só recebem a devida atenção após um evento grave, como aqueles que resultam em danos para o próprio utente. No entanto, os erros de identificação não se limitam a um fluxo de trabalho específico de saúde, ou a eventos que causam danos permanentes ou que promovam uma intervenção que sustente a vida. Estes podem ocorrer em qualquer cenário de saúde, incluindo cuidados ambulatoriais, internamentos, pedidos e administração de medicamentos e processos financeiros. A identificação incorreta do utente tem também um impacto no financiamento do hospital, porque os parceiros subsidiários recusam-se a pagar procedimentos médicos mal identificados [79].

Em 2007, Cruz-Correia et al. [56] descobriram uma omissão-chave na revisão sistemática onde a maioria das publicações não mencionou nenhum tipo de erros. Admitindo a

necessidade de verificar a qualidade dos dados integrados, para que ao invés de propagar erros de dados, emitir alertas sobre a qualidade dos mesmos para que possam ser acionados processos de correção [61], [80].

1.4.8. Trabalhos Relacionados

Em termos de investigação dos SI na saúde, existem alguns trabalhos que deram passos necessários para uma integração de dados mais coesa, onde a interoperabilidade é a chave, quando há partilha de dados dos utentes, referindo: [61], [80]–[83].

Em termos económicos, segundo o relatório *Global Healthcare IT Integration Market Growth, Trends & Forecasts* (2016-2021) [6], o mercado global de integração de TI na saúde é estimado em US \$2,35 mil milhões em 2016 e deverá atingir US \$3,92 mil milhões até 2021. Os programas de iniciativas e incentivos lançados pelos vários governos, o aumento da necessidade de reduzir os custos de cuidados de saúde, a crescente necessidade de informações de utentes de melhor qualidade, incluindo informações demográficas, clínicas e financeiras e necessidade de aumentar a eficiência das práticas médicas e das instituições, são os principais impulsionadores desse mercado global de integração de TI na saúde. Além disso, o alto retorno dos investimentos, no serviço digital, têm impulsionado o mercado. No entanto, a tecnologia também tem barreiras a serem superadas, tal como o alto custo dos investimentos iniciais, a falta de profissionais de TI adequadamente com formação em saúde e vários problemas de interoperabilidade entre os diferentes produtos de TI de saúde. A integridade e a segurança dos dados também são uma preocupação fundamental [82], [83].

É de salientar que o uso de dados clínicos na investigação médica tem um enorme potencial de retorno no conhecimento [80], [81], mas também na angariação de fundos para as instituições de saúde. No entanto, vários autores confirmam que garantir a segurança e a privacidade dos dados, superar os desafios associados à integração de diversos sistemas e manter a infraestrutura para o uso de dados de alta qualidade são hoje alguns dos principais desafios associados ao uso de EHR na investigação clínica [7].

Como consequência da investigação e da revisão bibliográfica efetuada, não existem muitos artigos que façam a análise do nome de utentes. No entanto, a Datavant é uma empresa firmemente comprometida com a evolução na saúde através da análise de dados. Esta empresa criou um algoritmo fonético SOUNDEX que é também um codificador de referência. SOUNDEX é um algoritmo que codifica similarmente sons idênticos de nomes (em inglês) como um valor consistente [70]. SOUNDEX é comumente usado quando os apelidos correspondentes em conjuntos de dados como variações na ortografia são comuns

na entrada de dados. Cada código SOUNDEX gerado a partir de uma cadeia de caracteres de texto de entrada tem 4 caracteres – a primeira letra do nome e, em seguida, 3 dígitos gerados a partir dos caracteres restantes, com elementos fonéticos de sonoridade semelhante codificados da mesma cadeia (por exemplo, D e T são ambos codificados como um valor 3, M e N são ambos codificados como um 5) [70]–[72].



Sistema de Validação de Nomes de Utentes

2. Sistema de Validação de Nomes de Utentes

2.1. Desenho do Estudo

Neste capítulo, descreve-se a população alvo, a amostra, as estratégias de recolha de dados, bem como, a metodologia de análise dos dados realizada do projeto. Descrevem-se as variáveis ou grupos de sujeitos, examinam-se as relações de associação entre as mesmas, assegurando o seu controlo. As variáveis permitem explicar e reduzir fontes de erro das mensagens partilhadas no sistema de saúde. Uma das variáveis associada aos dados dos utentes é o nome, sendo testada por algoritmo para prever erros nas mensagens. Os elementos do desenho do estudo seguiram o referido em Marie-Fabienne Fortin [15], [84].

Como população alvo deste projeto tem-se os SI, de saúde utilizados nas instituições hospitalares nacionais. A amostra contempla os SI existentes nas instituições de saúde envolvidas no projeto: *Instituição 1 e Instituição 2*. Opta-se pelo anonimato das instituições de saúde por questões de ética, explicada num dos pontos seguintes nesta dissertação.

Na Tabela 1, apresenta-se uma lista dos SI, com o respetivo tipo de serviço. Os serviços em cada uma das instituições de saúde vão ser alvo de estudo, podendo beneficiar da implementação do HS.HELIOS.

Tabela 1: Sistemas de informação de saúde instalados nas duas instituições.

TIPO DE SISTEMA	INSTITUIÇÃO 1	INSTITUIÇÃO 2
Gestão de utentes	SONHO (SPMS)	SONHO (SPMS)
Urgência	Medtrix (interno)	Alert (Alert)
Urgência armários medicação	Supply Point (BIQ) Integração prescrição e registo dispensa por utente/ enfermeiro/ serviço HL7	n/a
Urgência triagem	Sclinico (SPMS)	Alert (Alert)
Urgência pediátrica triagem	Triagem canadiana (BIQHS)	Alert (Alert)
Urgência - arquivo digital	BhealthECG (BYME) ECG	n/a

TIPO DE SISTEMA	INSTITUIÇÃO 1	INSTITUIÇÃO 2
Consulta	Medtrix (interno)	Alert (Alert)
Internamento	Medtrix (interno)	Sclinico (SPMS)
Smip (ucip / uci)	B-ICUCARE (B-simple)	n/a
Bloco / anestesia	B-Anesthetic (B-simple)	n/a
Obstetria	ObsCare (VirtualCare)	Sclinico (SPMS)
ICU e anestesia	BICUCare (BSimple)	n/a
Serviço de sangue - imuno	HYT	Sibas (Glintt)
Serviço de sangue	GOTA (Weiber)	Sibas (Glintt)
Lab. de patologia clínica	Appolo (Confidena)	Modulab (Werfen)
Lab. Anatomia patológica	CITOPRO (AMBIDATA)	AnaPat (Glintt)
Radiologia	BHealthFlow (BYME)	Radio (Glintt)
Pacs	Sectra PACS (Sectra)	(Siemens)
Exames especiais (orl / oftal / obsgino / pneumo / gastro)	Bhealthflow e Bhealth sHOT (Byme)	Gastro - Radio / (Glintt)
Prescrição interna	Medtrix (interno)	SGICM (Glintt)
Prescrição externa	PEM (SPMS)	PEM (SPMS)
Gestão de farmácia	Ghaf (Glintt)	SGICM (Glintt)
Registos de enfermagem	Medtrix (interno)	SGICM (Glintt)
Registos enfermagem - administração fármacos	Medtrix (interno)	SGICM (Glintt)
Alimentação utentes	NUTRIX (integração com MEDTRIX dietas)	SGICM (Glintt)
Pds - processo clínico	WEBMEDTRIX (interno)	Sclinico (SPMS)
Teleradiologia	GS24 (HL7) + ITM (WS vai ser descontinuado)	Campos Costa (Integração direta com o RADIO no envio e receção de relatórios)
Telemedicina	Campos Costa + LENITUDES (HL7) Execução MCDT exterior(Ressonância magnética / Medicina Nuclear/Molecular) Joaquim Chaves Execução MCDT Patologia Clínica (HL7)	Cardiologia Pediátrica - Hospital Pediátrico Medigraf (ferramenta descontinuada)
Codificação clínica	WEBGDH (SPMS)	WEBGDH (SPMS)

Entre os SI de saúde destacam-se alguns com as seguintes características:

- SONHO (SPMS): Sistema CORE de articulação entre as várias aplicações na instituição. Apoio ao processo administrativo e financeiro hospitalar.
- Alert (Alert): Receção, triagem e resposta aos pedidos de agendamento das consultas hospitalares, emitidos por profissionais dos Cuidados de Saúde Primários do SNS.
- Sclinico (SPMS): Acesso e registo de informação no EHR do utente, em consulta e internamento, por médicos e enfermeiros. Este possui integração de várias aplicações.

- AnaPat (Glintt): Comunicação e acesso a relatórios de exames histológicos de biópsias.
- Radio (Glintt): Agrega e permite o acesso aos resultados dos exames de imagem. Tem integração com o visualizador de imagens e leitor de relatórios.
- SGICM (Glintt): Sistema de apoio à prescrição medicamentosa do utente internado, pedido à farmácia hospitalar, integração com dispensadores automáticos e registo de consumos.
- PEM (SPMS): A PEM é a aplicação de prescrição de medicamentos para ambulatório. É uma aplicação *Web* e utiliza tecnologia *Java* para interface com o *middleware* do Cartão de Cidadão ou da Ordem dos Médicos no processo de autenticação na plataforma. Permite a prescrição de medicamentos listados na base de dados *Infomed*, privilegiando a denominação comum internacional do princípio ativo (face ao nome comercial).

Existem outros sistemas mais especializados, bem como aparelhos médicos (como os sexplos: ecógrafos e ECGs.) que podem beneficiar de um sistema central de integração como o HS.HELIOS. Para o caso do HS.HELIOS, os critérios de inclusão e exclusão são:

Inclusão:

- SIs que transmitam dados entre os Centros Hospitalares considerada relevante para o estudo da partilha de mensagens.

Exclusão:

- SIs descontinuados.
- SIs que não tenham contrato de manutenção.

Para o presente estudo, nas instituições analisadas, identificaram-se 53 SI. A representatividade da amostra reproduz os mesmos SI existentes nas instituições de saúde estudadas. Portanto, obteve-se o mesmo grau de representatividade, com os SI que respondem aos critérios de inclusão. No tamanho da amostra estão incluídos os mesmos SI instalados nos Centros Hospitalares, uma vez que se pretende a extração de métricas relacionadas com a troca de mensagens HL7 nos dois sentidos da transmissão. Desta forma, analisam-se os sistemas de forma independente e de forma integrada.

2.2. Instrumento de recolha de dados

Para esta investigação, as mensagens analisadas fazem parte das instituições de saúde portuguesas, propositadamente anonimizadas. Estas possuem características equiparáveis ao nível dos SI utilizados e dimensão dos serviços dos departamentos analisados. As mensagens serão obtidas a partir de uma aplicação específica desenvolvida pela HealthySystems, Lda. que permite a recolha e tratamento das mensagens HL7 dos diversos canais, presentes nas unidades hospitalares.

Para apoiar a monitorização da rastreabilidade da informação, foram desenvolvidos *dashboards* específicos com base na informação recolhida pelo integrador HS.HELIOS. Este projeto utiliza o barramento de integração de dados HS.HELIOS instalado no Serviço de Informática num hospital. Faz-se a análise da validação dos nomes dos utentes efetuados entre SIS, quantificando os que são bem sucedidos e os que são mal sucedidos, permitindo aferir o nível da eficácia e o nível da eficiência da interoperabilidade, para comparar ambientes hospitalares semelhantes. Além disso, confere-se o tempo de deteção dos pedidos entre sistemas, que não são bem sucedidos, verificando o tempo de espera do profissional de saúde a obter a informação desejada nos casos das mensagens bem sucedidas e nos casos em que são mal sucedidas.

2.3. Metodologia de análise dos dados

Para realizar a análise de dados foram utilizadas as métricas relacionadas com as integrações. As métricas foram quantitativas foram examinadas, registando-se sempre que identificadas as falhas e ou erros nas mensagens que identificam dados dos utentes. O rigor na identificação do detalhe métrico assenta na fidelidade e validade interna (exatidão dos resultados) e externa (generalização dos resultados). Assim, permite verificar se os resultados das falhas e erros valem para as diversas situações, para as duas instituições estudadas.

Para as métricas quantitativas os procedimentos foram:

- Número de mensagens HL7 trocadas com falhas no nome de utente na transmissão num sistema hospitalar.
- Percentagem de mensagens HL7 com falhas no total de mensagens trocadas.
- Tempo médio de transmissão de mensagens entre dois sistemas hospitalares.

Utilizaram-se as métricas relacionadas com as integrações. Procedeu-se à verificação da integridade semântica das mensagens HL7 trocadas e a monitorização em tempo real das integrações.

Para o caso da rastreabilidade da informação e monitorização em tempo real das integrações, foi identificado o percurso das mensagens HL7, e foram criadas condições para se desenvolverem mecanismos (canais) opcionais para direccionar essas mensagens para um caminho mais adequado.

A verificação da integridade semântica efetuada através da validação do nome do utente e das mensagens HL7 trocadas, foi analisada por comparação do número de mensagens com erros antes e depois da implementação da primeira e segunda iteração do sistema. Realizou-se um estudo de forma a identificar a origem das anomalias, a fim de futuramente se tomar medidas para a sua resolução. Privilegiou-se a veracidade, solidez e a certeza.

O tempo de identificação do problema e o tempo de reposição das mensagens falhadas, nos casos das integrações mal sucedidas, foi quantificado num valor numérico para se poder calcular o tempo de resolução destes problemas.

3. Métodos

Os métodos para esta investigação foram baseados na bibliografia analisada e referida no estado de arte deste documento, acarretando as novidades que são relativas às variáveis selecionadas e as respetivas relações, como os exemplos dos softwares, as mensagens e os nomes dos utentes. Daí, foi necessário seguir uma sequência de tarefas, as quais passam a ser descritas:

1. **Levantamento dos sistemas existentes e construção de canais:** realizaram-se os levantamentos de integrações necessárias, e a criação dos canais de comunicação.
 - 1.1 Levantamento dos sistemas existentes: efetuado um levantamento de alguns dos SI mais comuns nos hospitais.
 - 1.2 Classificação dos sistemas quanto ao tipo de serviço hospitalar: classificaram-se os SI.
 - 1.3 Mapeamento dos sistemas com os respetivos domínios: mapear os SI.
 - 1.4 Seleção dos perfis a estudar: o promotor responsável identifica junto dos co-promotores os sistemas integrados no HS.HELIOS de acordo com a criticidade dos mesmos.
 - 1.5 Construção de canais para implementar perfis: o promotor responsável HLTSYS avança para a criação dos canais para integrações dos sistemas instalados nos co-promotores em colaboração com o centro hospitalar e seus fornecedores de soluções, permitindo assim integrações de sistemas no centro hospitalar.
2. **Levantamento da informação necessária a recolher no âmbito do Regulamento Geral sobre a Proteção de Dados (RGPD):** apuramento da informação no que diz respeito à necessidade de rastreabilidade e de portabilidade.
 - 2.1 Adaptação dos canais para recolher a informação: o promotor responsável entra em contacto com os fornecedores dos sistemas para definir os métodos de acesso e solicitar as alterações necessárias aos formatos das integrações e proceder-se-á a respetiva adaptação dos canais.
3. **Desenvolvimento da Validação de Nomes dos Utesntes (compatível com RGPD):** desenvolvimento de uma nova versão do HS.HELIOS que estenda a existente para conter a nova funcionalidade.
 - 3.1 Inquirir DPOs sobre a forma ideal de visualização dos resultados: os utilizadores são questionados sobre o modo que pensam ser ideal para a apresentação do fluxo de informação.

- 3.2 Criação de um *dashboard* de apoio à verificação das condições da RGD: realizou-se um levantamento de toda a informação já recolhida pelo HS.HELIOS e a sua relevância para RGD assim como analisada a necessidade de adicionar novas fontes de dados.

3.1. Ética

O projeto envolve o desenvolvimento de *software* para a centralização das transmissões de dados clínicos e o estudo da análise do impacto da implementação ao nível da redução dos tempos associados à prestação de cuidados de saúde, e de redução de erros na transmissão das mensagens. Desta forma, a utilização de dados clínicos dos utentes das duas entidades envolvidas no estudo, já contemplam os pedidos de ética necessários às respetivas comissões. Os dados dos utentes são os mesmos do projeto HS.HELIOS. Por outro lado, criaram-se canais de anonimato capazes de remover qualquer tipo de dados confidenciais e proceder ao envio de mensagens HL7 anónimas para um determinado conjunto de destinos diferentes, garantindo assim que estes destinos não recebem qualquer tipo de dados confidenciais de utentes. Assim, reforça-se que os dados para este trabalho já possuem de forma automática a autorização de comissões de ética. Para além disso, muita de informação a que se teve acesso vai estar associada a dados de fluxo de trabalho e de funcionamento de especialidades e não tanto a dados clínicos pessoais dos utentes. Os hospitais que fazem parte desta investigação ficam em anonimato. Da mesma forma, garante-se que os dados dos utentes, como o caso do nome, não estão explanados de forma a que sejam possível identificar qualquer cidadão. Assim, garante-se o recomendado sobre ética na saúde na bibliografia [15], [84].

O trabalho é descrito em três fases principais: o processo de recolha de dados, a descrição do algoritmo que avalia a qualidade do nome do utente e a sua implementação, e a forma como o sistema foi testado. A análise da frequência efetuação por iteração às variáveis com os dados do nome do utente.

3.2. Recolha e transformação de dados

As instituições de saúde precisam de soluções para partilhar informação entre os diferentes sistemas de forma fiável, segura e fácil de entender. Nesse sentido, a comunicação entre os sistemas de informação em saúde pode ter um intermediário que recebe as mensagens das aplicações de origem e as envie para as aplicações de destino. Esta solução intermediária tem os mesmos princípios que um *Enterprise Service BUS* (ESB) e trabalha como um mediador que promove o aumento da interoperabilidade e a redução dos erros de comunicação [85].

O cenário de interoperabilidade da instituição de saúde onde a solução foi testada, baseia-se no conceito de integração horizontal. A instituição de saúde possui um ESB, designado HS. HELIOS, que é responsável por monitorizar, em produção, as integrações entre os diferentes SI em saúde [86]. Este serviço é representado a cinzento na Figura 3.1. A arquitetura geral desta solução apresentada dentro das linhas azuis, visualiza, a utilização de um sistema ESB existente e operacional instalado numa instituição de saúde para extração de dados e posterior análise. Os dados enviados pela instituição de saúde ESB para a solução, consistem em eventos que contêm informações demográficas, mais especificamente a identificação do utente. Esses eventos são transmitidos pela rede de instituições de saúde entre os diferentes SI em saúde, e todos passam pela ESB, que reconhece o evento, duplica e os envia para análise. O formato dos eventos é baseado no padrão HL7 v2.x. A arquitetura da solução consiste em:

- três servidores;
- o processador;
- o servidor Web;
- a base de dados.

O servidor do processador é responsável por receber os dados e pode ou não transformá-los conforme seja necessário. Posteriormente, ele executa os dados através do algoritmo descrito neste trabalho, validando a qualidade do nome de utentes. O servidor WebServer possui uma API (*Application Programming Interface*) que oferecerá métodos para armazenar os resultados do algoritmo no servidor da base de dados e também fornece métodos para preencher um painel (*dashboard*) de fácil utilização, projetado para análise dos resultados do algoritmo. O servidor da base de dados é responsável por armazenar os dados que são analisados, bem como os resultados do algoritmo. Numa visão mais técnica, a

solução possui um mecanismo de integração de código aberto, *NextGen Connect* v 3.7.0, um servidor *Web Apache Tomcat* 8 e uma base de dados 10.3.9 relacional *MariaDB*.

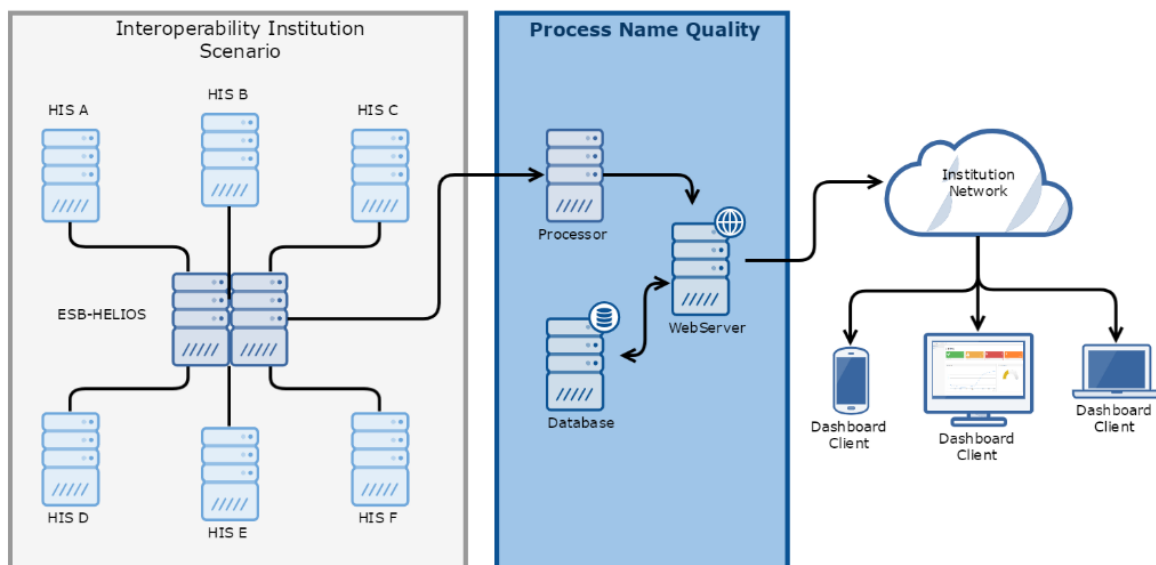


Figura 3.1: Arquitetura geral do sistema de *HS.HELIOS System*.

O fluxo de trabalho onde os dados circulam desde a rede da instituição até ao painel de fácil interpretação (*dashboard*) onde são visualizados os resultados do algoritmo é representado à direita na Figura 3.1.

3.3. Descrição do algoritmo

A cada mensagem HL7 trocada entre os SI de saúde através da ESB, foi extraído o nome do utente e enviado para validação. O sistema HS.HELIOS encaminhou as mensagens para o servidor de processamento da solução, que recebe os dados e executa o algoritmo. O algoritmo contém um conjunto de variáveis que permitem configurar a solução e estabelecer valores de referência.

Uma lista com os primeiros nomes atribuídos entre 2014 e 2017 publicada pelo Instituto de Registos e do Notariado de Portugal é utilizada como valor de referência, denominada Lista de Nomes Legais (LNL). Além disso, também é usada como valor de referência uma Lista de Nomes Inválidos (LNI). A lista LNI inclui palavras que foram identificadas como erros técnicos comuns e não podem fazer parte do nome de um cidadão português, como por exemplo:

- </PID.5.1>
- <PID.5.1>
- Bioquímica
- BK
- Bolsa
- Câmara
- Cardíaco
- Carro
- CC
- Controlo
- Desconhecido
- Doente
- Fem
- Fictício
- Filho
- Fule
- Funcionário
- Funcionários
- Geral
- Glicada
- Hemoglobina
- HLA-B27
- Identificado
- Indefinido
- Informática
- Integração
- Microbiologia
- Não
- Nascimento
- NEQAS
- Nulo
- Nutrição
- Oftalmologia
- Parentérica
- Rafafilho
- Rastreio
- Rastreio
- RNF

- RNM
- SEG
- Teste
- UCIP
- UK
- Urgência
- Utene

O algoritmo, procura evidências de 12 problemas diferentes regularmente encontrados, que são os seguintes:

- (1) o nome do utente está vazio;
- (2) o primeiro nome não está no LNL;
- (3) o primeiro nome tem menos de um número de caracteres;
- (4) o sobrenome tem menos de um número de caracteres;
- (5) o nome contém uma palavra com apenas uma consoante (situação específica para um nome Português);
- (6) o nome tem palavras na lista LNI (a pesquisa ignora os acentos);
- (7) o nome tem menos de um determinado número de palavras;
- (8) o nome tem um hífen que não está entre duas palavras;
- (9) o nome começa com um espaço;
- (10) o nome termina com um espaço;
- (11) o nome contém caracteres inválidos (caracteres considerados válidos são letras, letras com acentos ou caracteres configurados pelo usuário técnico do sistema);
- (12) o nome tem espaços contíguos.

Foi atribuído um peso a cada problema anteriormente referido, que pode ser configurado pelo utilizador do sistema através de um ficheiro de configuração (CF). O nível de qualidade é calculado a partir do complemento da soma desses pesos, como se apresenta na Equação 1.

$$\text{Nível de Qualidade} = 100\% - \sum_{i=0}^n (\text{peso do problema detetado}_i) \quad (1)$$

No que concerne ao problema em que o primeiro nome não está na lista LNL, o utilizador do sistema pode configurar se deseja validar o nome com acentos ou se os acentos são

ignorados. Os caracteres válidos também podem ser configurados pelo utilizador e pelos testes realizados, tendo sido considerados os seguintes caracteres válidos, (' e -), para cobrir casos como os nomes “O’Brien” ou “Castel-Branco”.

O algoritmo retorna um resultado que indica o nível de qualidade e a descrição dos problemas para um determinado nome de utente.

3.4. Avaliação da qualidade dos nomes dos utentes nas mensagens

A solução descrita nesta dissertação esteve em produção numa instituição de saúde portuguesa durante um período de 30 dias, em que 1.048.576 mensagens HL7 foram trocadas entre os sistemas e encaminhadas para o servidor de processamento. Destas mensagens a mesma quantidade de nomes do utente foi extraída, do campo do nome do utente (PID-5) das mensagens HL7 e validado pelo algoritmo.

Os dados recolhidos foram validados em duas iterações. A primeira iteração em tempo real e a segunda iteração após a atualização dos valores de referência. Após a conclusão da primeira iteração, realizou-se uma análise dos problemas detetados que levaram à atualização das listas de LNL e LNI.

Em alguns nomes que o algoritmo não identificou como nomes corretos, porque eles não estavam na lista LNL, embora houvesse um nome de correspondência de género na lista. Em seguida apresentam-se alguns exemplos dessas correspondências, em que a primeira coluna contém o nome não identificado e a segunda exibe um nome pertencente à lista LNL:

- Adílio **Adília**
- Aguinaldo **Aguinalda**
- Albertino **Albertina**
- Albina **Albino**
- Alício **Alícia**
- Almerindo **Almerinda**
- Almira **Almiro**
- Altina **Altino**
- Álvaro **Álvaro**

- Alvarina **Alvarino**
- Arcelina **Arcelino**
- Armandina **Armandino**
- Arménia **Arménio**
- Arminda **Armando**
- Belarmina **Belarmino**
- Bernarda **Bernardo**
- Bernardino **Bernardina**
- Carmindo **Carminda**
- Celestina **Celestino**
- Cesária **Cesário**
- Cira **Ciro**
- Cirilo **Cirila**
- Claudina **Claudino**
- Constâncio **Constância**
- Custódia **Custódio**
- Esmeraldo **Esmeralda**
- Fausta **Fausto**
- Feliciano **Feliciano**
- Felismina **Felismino**
- Firmina **Firmino**
- Florêncio **Florência**
- Florentino **Florentina**
- Florindo **Florinda**
- Francelino **Francelina**
- Gilberta **Gilberto**
- Guilhermino **Guilhermina**
- Hermínia **Hermínio**
- Inácia **Inácio**
- Isabelino **Isabelina**

- Josefino **Josefina**
- Justina **Justino**
- Leonídia **Leonídio**
- Leonilda **Leonildo**
- Marciano **Marciana**
- Marcília **Marcílio**
- Marinho **Marinha**
- Maurícia **Maurício**
- Máxima **Máximo**
- Maximiana **Maximiano**
- Natalino **Natalina**
- Natálio **Natália**
- Norberta **Norberto**
- Olindo **Olinda**
- Paulino **Paulina**
- Rogéria **Rogério**
- Silvano **Silvana**
- Silvério **Silvéria**
- Valdemira **Valdemiro**
- Verdiana **Verdiano**
- Virgília **Virgílio**
- Vitalina **Vitalino**

Estes nomes foram adicionados à lista LNL. A atualização da lista LNI foi realizada através da análise dos nomes que tinham um nível de qualidade inferior e palavras que não podiam pertencer ao nome de um cidadão, como apenas o primeiro nome ou os seguintes exemplos:

- Ângelo
- DOENTE FICTICIO TESTE INFORMATICA
- FUNCIONARIOS FUNCIONARIOS

- RNF (...)
- RNM (...)
- FUNCIONARIO FUNCIONARIO
- CONTROLO BOLSA NUTRICAO PARENTERICA
- RASTREIO OFTALMOLOGIA
- funcionário
- SEG RNF (...)
- PRIM RNF (...)
- DOENTE TESTE FICTICIO
- Doente de Teste 02
- RNF CATIA PATRICIA CAMPOS OLIVEIRA GUEDES
- MASC NAO IDENTIFICADO
- MASC IDENTIFICADO NAO
- FULE CAMARA FATY
- SEG RNM (...)
- JO RNM (...)
- CC (...)
- UK NEQAS MICROBIOLOGIA
- FEM NAO IDENTIFICADO
- Fem Identificado Nao
- cardiologia
- DOENTE INFORMATICA FICTICIO TESTE
- MAMADI CAMARA
- UK NEQAS MICROBIOLOGIA - BK
- UK NEQAS CARDÍACO 2018
- ASTOLPHO FILHO BARSOTELLI
- UK NEQAS BIOQUÍMICA GERAL 6000
- UK NEQAS HEMOGLOBINA GLICADA 2018
- UK NEQAS BIOAQUIMICA GERAL 8000 2018
- DOENTE UCIP FICTICIO

- FILHO (...)
- MAIM RNF (...)
- UK NEQAS HLA-B27
- ana
- UTENE TESTE INTEGRACAO
- de carro urgência
- PRIM RNF (...)
- NEQAS CONTROLO QUALIDADE IMUNO
- PRIM RNF (...)
- SEG FILHO (...)
- RAG FILHO (...)
- RAFAFILHO (...)
- PRIM RNF (...)
- PRIM RNF
- PRIM
- SEG RNM (...)
- SEG RNM
- SEG
- Cardiologia;
- < \PID.5.1 >;
- Doente;
- Microbiologia.

As duas iterações foram executadas com duas configurações diferentes na solução. O utilizador configurou a solução para um mínimo de duas palavras, cenário M2P; ou três palavras, cenário M3P, para o nome completo do utente.

4. Resultados

Através da análise de impacto da implementação do HS.HELIOS e das métricas utilizadas ao nível da redução de erros na transmissão de mensagens e redução dos tempos associados à prestação de cuidados de saúde demonstra-se que, a implementação do projeto numa instituição de saúde permite a obtenção dos seguintes resultados:

1. **Qualidade de dados** - a falta de qualidade e a propagação de dados com erros entre SI dificulta a análise e a gestão das unidades hospitalares. Muitas das comunicações entre sistemas têm erros que dificilmente os utilizadores ou os técnicos de informática detetam. O HS.HELIOS deteta erros nos dados trocados entre sistemas automaticamente para posteriormente serem alertados os serviços adequados.
2. **Rastreabilidade da informação** - podendo ser definida como a capacidade de traçar o caminho desde a geração, comunicação e uso de um documento ou mensagem clínica. O estudo pretende permitir saber, através de um identificador, qual o fluxo dessa informação, detetando acessos suspeitos, bem como avalia a utilização de um canal em particular.
3. **Gestão e monitorização de processos** - os fluxos de informação dentro dos hospitais têm um impacto muito significativo na capacidade de produção. Como a dificuldade de comunicação de resultados de exames, ou o atraso no envio de requisições implica atrasos no atendimento de utentes e eventualmente nas altas hospitalares. O *software* implementado possibilita conhecer os fluxos de informação para que possam ser comparados e adaptados aos processos de trabalho que a instituição pretende promover.
4. **Suporte à operação do Serviço de informática** - os centros hospitalares são hoje instituições muito dependentes dos inúmeros SI. Esta investigação identifica a origem de problemas de transmissão de dados. Assim, com a aplicação proposta, reduzem-se os tempos de espera e do número de falhas.

A qualidade das integrações em termos de completude dos campos requeridos e condicionais é semelhante nas instituições analisadas.

Relativamente ao algoritmo de validação de nomes dos utentes, na iteração 1 e nos dois cenários, 2.384 (0,2%) nomes dos utentes foram encontrados com palavras inválidas que pertenciam à lista de LNI. Para o cenário M2P, foram detetados 9.905 (0,1%) nomes de utentes completos e incorretos, entre os quais 2.701 (27%) tinham apenas uma palavra. Foram detetados 40.699 (3,9%) nomes incorretos no cenário M3P, entre os quais 33.257 (82%) eram nomes com menos de 3 palavras (Tabela 2).

Tabela 2: Frequência e percentagem para a análise LNL e LNI num total de 1 048 576 mensagens.

Descrição do problema	WM2 cenário				WM3 cenário			
	Iteração 1		Iteração 2		Iteração 1		Iteração 2	
	N	%	N	%	N	%	N	%
O nome não tem problemas, ignorando a acentuação	991 538	94,6%	993 345	94,7%	964 088	91,9%	965 847	92,1%
O primeiro nome não está na LNL, ignorando a acentuação	995 081	94,9%	998 696	95,2%	995 081	94,9%	998 696	95,2%
O nome não tem problemas, considerando a acentuação	866 873	82,7%	871 814	83,1%	868 420	82,8%	873 146	83,3%
O primeiro nome não está na LNL, considerando a acentuação	861 538	82,6%	864 714	82,5%	861 538	82,6%	864 714	82,5%
O nome não tem palavras na lista LNI	2 384	0,002	7 931	0,007	2 384	0,002	7 931	0,007
Os nomes estão incorretos	9 905	0,1%	44 699	3,9%	9 905	0,1%	44 699	3,9%
Os nomes que o algoritmo deteta como incorretos	57 038	5,4%	584 488	8,1%	5 231	5,3%	82 729	7,9%

No início da primeira iteração, a lista LNL continha 7.472 nomes legais e a lista LNI continha 5 nomes inválidos. A fim de obter os elementos mais relevantes para o resultado pretendido, a frequência e a taxa de erro foram calculadas para os problemas apresentados, conforme necessário. Após a análise da iteração 1, 64 nomes foram extraídos para serem adicionados à lista LNL e 41 palavras a serem inseridas na lista LNI. Como consequência, para a segunda iteração, a lista LNL tem 7.536 nomes e a lista LNI contém 46 nomes.

Em relação ao primeiro problema de reconhecimento de nome usando a lista LNL, houve um pequeno crescimento na deteção de nomes corretos da iteração 1 para a iteração 2, como o mostrado na Tabela 2. Além disso, determinamos que os nomes corretos que o algoritmo determinou como incorretos diminuiu 3.615 (7,7%) no cenário M2P e 3.567 (8,1%) no cenário M3P. Não foram detetados nomes com hífen ou com espaços indevidos. Relativamente ao

problema dos nomes inválidos, verificou-se que a frequência aumentou muito da primeira iteração para a segunda, de 2.384 para 7.931 (56% das mensagens com erros para o cenário M2P e de 13,6% de mensagens com erros para o cenário M3P). Com a adição de apenas 46 nomes inválidos, foi possível detetar mais 5.547 nomes incorretos no conjunto de dados.

Tabela 3: Frequência e taxa de erro nas duas iterações num total de 1.048.576 mensagens.

Descrição do problema	Iteração 1 e 2	
	N	Taxa de erro
O nome tem menos de 3 palavras (cenário M3P)	33 257	31, 7‰
O nome tem menos de 2 palavras (cenário M2P)	2 701	2, 6‰
O nome contém caracteres inválidos	728	0,7‰
O nome do utente está vazio	428	0,4‰
O primeiro nome tem menos de um determinado número de caracteres	132	0, 13‰
O nome contém uma palavra com apenas uma consoante	19	0,02‰

Todos os nomes incorretos associados aos problemas listados na Tabela 3 foram encontrados. Como a deteção desses nomes não está vinculada ao LNL e ao LNI, os valores de frequência correspondentes permaneceram nas duas iterações.

A velocidade média de processamento foi calculada a fim de determinar o impacto do desempenho desse algoritmo na produção. O intervalo de tempo é em ordem de 1 ou 2 milissegundos para as características de servidor usadas.

O *dashboard* criado para visualizar a quantidade de nomes de utentes com erros detetados pode ser visualizado no SI, conforme representamos na Figura 4.1. A quantidade de nomes com erros pode também ser visualizada com a mesma tecnologia, veja-se a Figura 4.2.

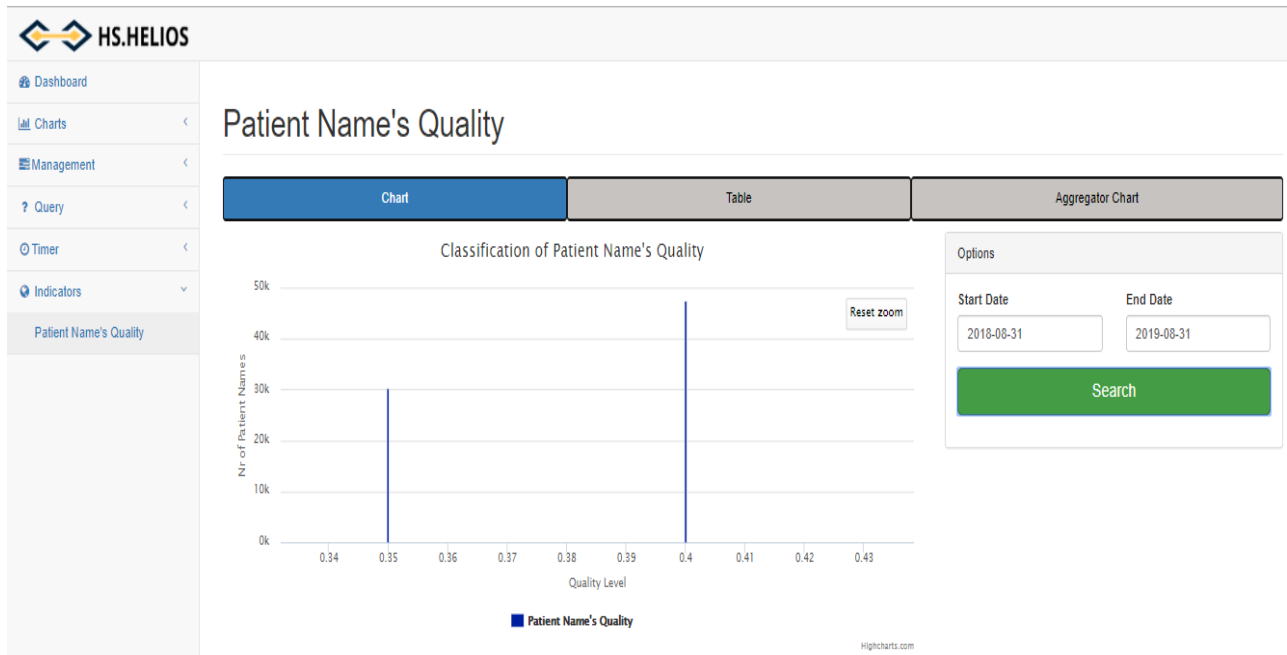


Figura 4.1: Apresentação do *dashboard* criado para visualização da quantidade de nomes detetados com erros.

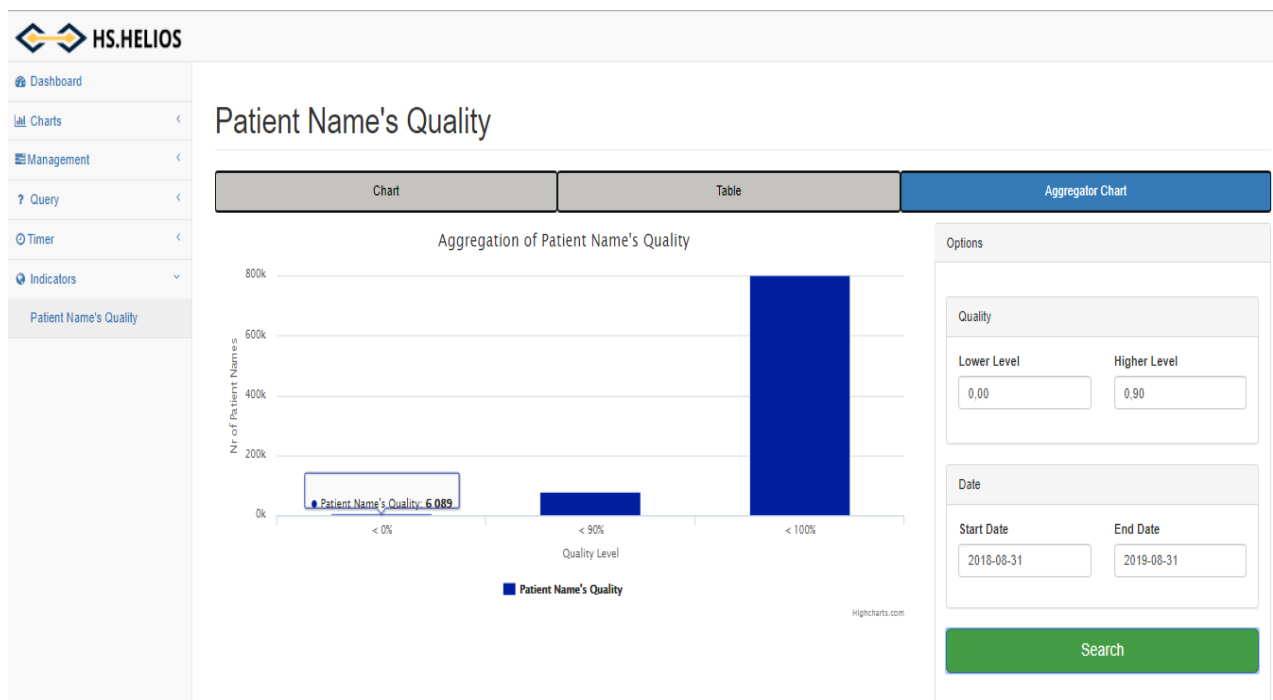


Figura 4.2: Apresentação do *dashboard* que visualiza a quantidade de nomes detetados com erros, agregados por nível de qualidade.

Como consequência da análise de resultados, foi necessário criar uma classe *checkName*, implementada na linguagem *Java* e compilada numa biblioteca a ser importada pela solução,

a fim de classificar os nomes dos utentes por tipos. A classe *checkName*, visualizada na Figura 4.3, consiste em 7 tipos de classificação de nomes, que são os seguintes: (i) *isOK* - nome com nível máximo de qualidade; (ii) *hasInvalidName* - nome que contém uma palavra inválida; (iii) *hasInvalidCharacters* - nome que contém caracteres inválidos; (iv) *unknownPatient* - nome de utente desconhecido, que pode estar vazio ou conter palavras que indicam que o utilizador que inseriu os dados do utente não tinha meios de identificar o utente, por exemplo, "nome do utente", "sem identificação", "macho não identificado", "fêmea não identificada"; (v) *newBorn* - nome que começa com "filho", "filha", "filho de" ou "filha de"; (vi) *isTest* - nome que contém palavras para realizar testes no sistema, por exemplo "teste", "informática", "qualidade", etc. e (vii) *isMarked* - nome que começa com "RNF", "RNM", "primeiro", "segundo", etc.

A classe *checkName* recebe o nome do utente e o método a ser executado e retorna um objeto *JSON* com o nome do utente, a pergunta feita e a resposta com um valor Verdadeiro ou Falso. O esquema do algoritmo representa-se na Figura 4.3.

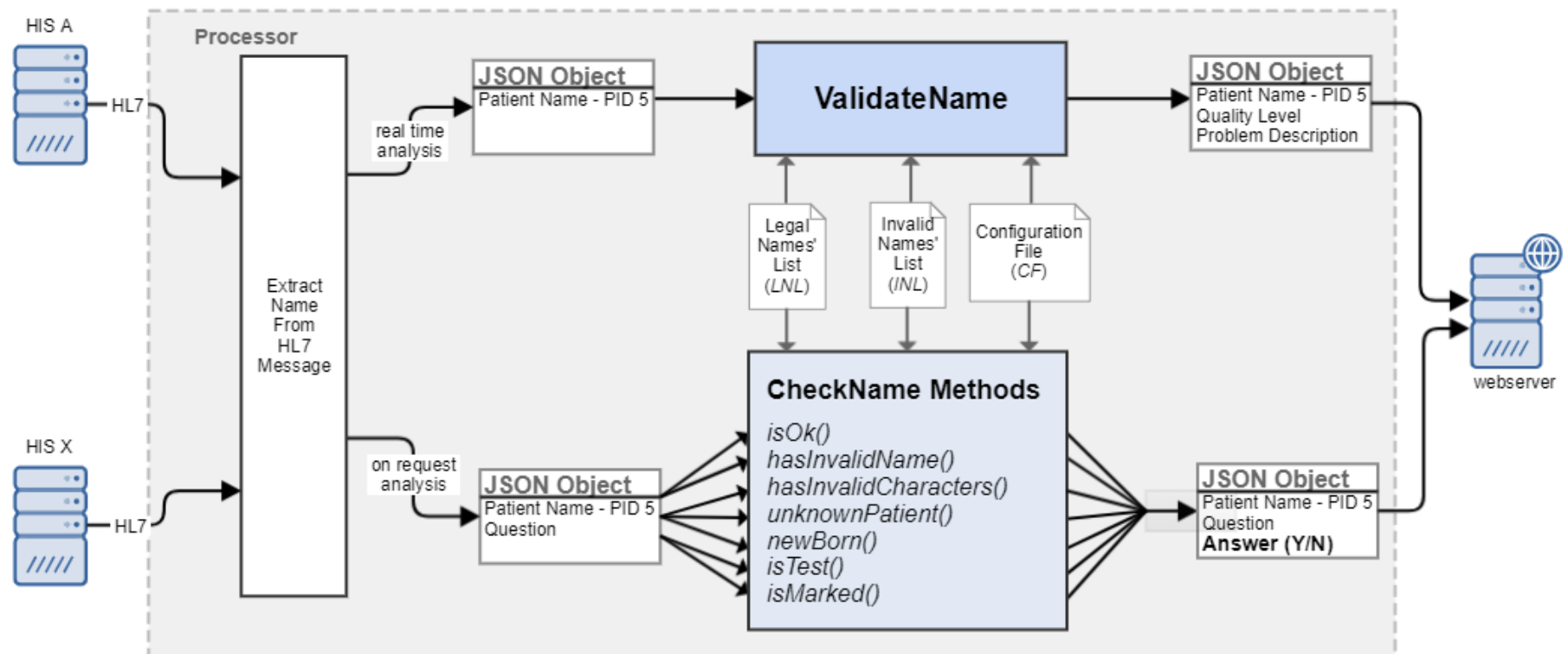


Figura 4.3: Esquema do algoritmo.



Discussão

5. Discussão

Ao monitorizar a troca de mensagens HL7 entre sistemas, podemos fornecer os meios para superar a heterogeneidade de dados desses sistemas e aumentar a sua qualidade. O crescimento da consciencialização e da visibilidade destas questões num ambiente institucional de saúde, possibilita a redução de problemas relacionados com falhas de identificação de utentes e aumenta a melhoria da qualidade dos dados.

A solução apresentada neste trabalho permitiu realizar um estudo sobre alguns erros mais encontrados nos nomes de utentes em hospitais portugueses. Este estudo também revelou situações sistemáticas que ocorreram entre dois ou mais sistemas informáticos de saúde. Uma dessas situações foi quando ao serem enviadas mensagens de um sistema de informação de saúde A apenas para um sistema de informação de saúde B, sempre era remetido um nome codificado, "Ângelo", no segundo campo do nome do utente. A instituição em causa foi notificada e demonstrou grande agrado ao prontamente se agir em conformidade para colmatar a falha.

Em relação aos nomes inválidos, o algoritmo revelou que, com algumas entradas adicionadas à lista LNI, um grande número de nomes incorretos foi detetado na segunda iteração dos dados. A maioria dos nomes adicionados à LNI veio da análise feita para os nomes processados na primeira iteração que apresentavam uma reduzida qualidade. Estes nomes de qualidade inferior consistiram principalmente em nomes para testes do sistema, nomes marcados para exames laboratoriais, erros de codificação, nomes de departamentos e informação de seguro, revelando a diversidade da informação que pode vir num campo do nome do doente, que conduz a uma identificação inadequada do utente. Mais grave ainda, será a propagação destes erros por vários sistemas numa instituição ou mesmo entre inúmeras instituições a nível nacional.

Com a solução instalada, o sistema atua como um integrador, um auditor padrão, um interpretador e uma fonte secundária de dados, nomeadamente para nomes de utentes, resolvendo alguns dos problemas de integração que afetam a Instituição Hospitalar. A ferramenta pode capacitar a Instituição com conhecimento suficiente sobre o Sistemas de Informação de Saúde, permitindo assim um melhor controlo sobre os serviços prestados

pelos fornecedores. No final, o utente é o mais afetado por estas falhas e, embora isso tenha custos financeiros para o HI, os fornecedores raramente são afetados.

Um dos principais desafios encontrados no desenvolvimento da nossa infraestrutura esteve relacionado com o acesso aos dados dos utentes e por este motivo este estudo ficou limitado apenas à amostra abordada e presente no nosso sistema. Em relação à arquitetura, inicialmente o estudo estatístico dos dados baseou-se na tecnologia do integrador *Mirth Connect®* e na base de dados *Maria DB*. Este processo mostrou-se muito moroso, demorando mesmo muitas horas de processamento. Por este motivo surgiu a necessidade da criação da aplicação desenvolvida na linguagem *Java*, cujo código fonte se apresenta no Anexo 8.2, e que numa média de 30 segundos executava o processamento de um milhão de nomes de utentes.

Os problemas de integração, se não forem detetados, podem originar o cancelamento de consultas, exames laboratoriais, cirurgias, perdas nas prescrições de medicamentos e até faturação indevida. Com a fracturação da informação, consequência de falhas na integração, leva a que apenas um sistema esteja ciente do procedimento. Estes problemas podem ser particularmente graves, uma vez que os utentes não são informados do cancelamento dos referidos procedimentos, fazendo com que visitem o HI na data anteriormente marcada, apenas para ser informado de que um erro nos sistemas causou o mal-entendido.

Alguns trabalhos estudaram os erros na saúde, referindo a vantagens de existir um alerta para evitar erros como a troca da medicação dos utentes [30]. Os conceitos e a relação entre as variáveis nas investigações é que nos permitem identificar as fontes de erros. Uma das diferenças entre a nossa investigação e de Atasoy et al [30] é a seleção da principal variável a analisar o nome dos utentes, para poder identificar as diferentes tipologias de erros nos diversos *softwares* utilizados, e efetuar um algoritmo para a sua correção.

Para atingir o sucesso, a fase de conceção do produto teve em conta questão dos erros nos nomes, onde foram concentrados esforços para envolver os utilizadores no processo e implementar esta fase em várias iterações.



Conclusões e Trabalho Futuro

6. Conclusões e Trabalho Futuro

6.1. Conclusões

Através da pesquisa realizada, foi detetado que o campo de nome é usado para outras funções que não aquela para a qual foi projetada. Este campo tem sido utilizado para realizar testes, marcar recém-nascidos, especialidades médicas ou outras marcações. Esse tipo de uso indevido pode causar falhas de sistema com consequências inesperadas e essas falhas podem ser propagadas para outros sistemas ou bases de dados. Por outro lado, a validação do nome dos utentes ajuda a manter a precisão na troca de mensagens entre sistemas.

Acreditamos que as questões associadas à infraestrutura referida aumentam grandemente o propósito da interoperabilidade dos cuidados de saúde e podem levar a desafios complexos de integração que não podem ser facilmente resolvidos sem a Intervenção dos fornecedores de software.

Desta forma, podemos concluir que o padrão HL7 necessita de ser usado corretamente e com o menor número de erros possível. Associada à infraestrutura proposta, a criação de mecanismos específicos capazes de analisar o conteúdo de cada mensagem HL7 abre caminho para o desenvolvimento de sistemas de alerta eficientes numa infraestrutura de interoperabilidade.

6.2. Trabalho Futuro

Para o trabalho futuro, foram detetados vários problemas na identificação dos utentes através dos seus nomes e foi sentida a necessidade de fazer o mesmo com todos os campos de identificação do utente de uma mensagem HL7. Consideramos que seria útil estender o

reconhecimento a mais do que o primeiro nome, o que implica a pesquisa e criação de uma extensa lista de palavras válidas para um nome de utente. Torna-se um desafio estender a lista LNL para nomes de estrangeiros. Para resolver este problema, o sistema irá permitir a um utilizador técnico decidir se -um determinado nome é aceitável. Nesse caso, o nome seria adicionado à referida lista LNL. Em termos de alarmística, quando um nome incorreto é detetado, a mensagem correspondente pode ser processada, encaminhada para um destino diferente ou criado um alarme para receber a intervenção de um utilizador técnico.

O desempenho de tais componentes é importante e pretende-se analisá-lo com posterior investigação. Para verificar se a relação estava a atuar de acordo com a especificação, é importante desenvolver-se um sistema capaz de disparar alarmes quando uma determinada atividade não segue o fluxo de trabalho descrito. Ainda mais visível: o sistema deve identificar muitos desvios do fluxo de trabalho da mensagem esperada e criar uma plataforma de auditoria em tempo real para as mensagens HL7 trocadas pelos fornecedores dentro de um HI.

Como as mensagens HL7 têm um padrão bem definido, podemos usar o mecanismo de integração no núcleo da infraestrutura para criar um conjunto de canais capazes de analisar o conteúdo de cada mensagem e acionar alertas se o padrão da mensagem não estiver a ser usado corretamente.

A ferramenta tecnológica poderá possibilitar a criação de mecanismos de monitorização mais precisos e úteis para os serviços das TI, que podem continuar a ser baseados nas mensagens HL7, e que têm algum tipo de controlo ou códigos de erro para avaliar se um determinado procedimento foi realmente bem sucedido ou bem interpretado pelo sistema final. Deseja-se que seja possível o fornecimento de informação em tempo real sobre os problemas existentes em sistemas críticos, que de outra forma são considerados como caixas pretas onde apenas os fornecedores de software podem aceder.



Referências

7. Referências

- [1] M. Noura, M. Atiquzzaman, e M. Gaedke, «Interoperability in Internet of Things: Taxonomies and Open Challenges», *Mob. Netw. Appl.*, vol. 24, n. 3, pp. 796–809, Jun. 2019.
- [2] C. Neves, «Qualidade em Saúde: Optimização de Sistemas de Informação», 2012. [Em linha]. Disponível em: https://repositorio.ul.pt/bitstream/10451/9464/1/ulfc105400_tm_Carina_Neves.pdf.
- [3] J. Adler-Milstein e A. Jha, «Systematic review of scope and quality of electronic patient record data in primary care», *Br. Med. J.*, vol. 326 (7398), p. 1070, 2003.
- [4] V. C. Emeakaro, P. Healy, K. Fatema, e J. P. Morrison, «Analysis of data interchange formats for interoperable and efficient data communication in clouds», apresentado na Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing, 2013, pp. 393–398.
- [5] R. Noumeir, «Active Learning of the HL7 Medical Standard», *J. Digit. Imaging*, vol. 32, n. 3, pp. 354–361, Jun. 2019.
- [6] X. Gansel, M. Mary, e A. van Belkum, «Semantic data interoperability, digital medicine, and e-health in infectious disease management: a review», *Eur. J. Clin. Microbiol. Infect. Dis.*, vol. 38, n. 6, pp. 1023–1034, Jun. 2019.
- [7] «A robust health data infrastructure», *Agency for Healthcare Research and Quality*, 2013. [Em linha]. Disponível em: https://www.healthit.gov/sites/default/files/ptp13-700hhs_white.pdf.
- [8] L. Zhang e X. Xu, «A community public health system design based on hl7 criterions», *Comput. Inf. Sci.*, vol. 4, n. 2, p. 148, 2011.
- [9] E. Hatef, J. P. Weiner, e H. Kharrazi, «A public health perspective on using electronic health records to address social determinants of health: The potential for a national system of local community health records in the United States», *Int. J. Med. Inf.*, vol. 124, pp. 86–89, 2019.
- [10] J. R. Camacho, S. Stäubert, e M. Löbe, «Automated Import of Clinical Data from HL7 Messages into OpenClinica and tranSMART Using Mirth Connect.», *Stud. Health Technol. Inform.*, vol. 228, pp. 317–321, 2016.
- [11] M. C. Weinstein, L. B. Russell, M. R. Gold, e J. E. Siegel, *Cost-effectiveness in health and medicine*. Oxford university press, 1996.
- [12] E. E. Zusman, «Meeting meaningful use objectives for Electronic Health Record implementation», *Neurosurgery*, vol. 69, n. 2, pp. N24–N26, 2011.
- [13] L. Sprague, «Meaningful use of health information technology: proving its worth?», 2015.
- [14] C. S. Wood *et al.*, «Taking connected mobile-health diagnostics of infectious diseases to the field», *Nature*, vol. 566, n. 7745, pp. 467–474, Fev. 2019.
- [15] M.-F. Fortin, *Fundamentos e etapas do processo de investigação*. Lusodidactica, 2009.
- [16] D. Blumenthal e M. Tavenner, «The “meaningful use” regulation for electronic health records», *N. Engl. J. Med.*, vol. 363, n. 6, pp. 501–504, 2010.

- [17] A. K. Jha, «Meaningful use of electronic health records: the road ahead», *Jama*, vol. 304, n. 15, pp. 1709–1710, 2010.
- [18] S. S. Jones, R. S. Rudin, T. Perry, e P. G. Shekelle, «Health information technology: an updated systematic review with a focus on meaningful use», *Ann. Intern. Med.*, vol. 160, n. 1, pp. 48–54, 2014.
- [19] N. Sood, A. Ghosh, e J. Escarse, «The Effect of Health Care Cost Growth on the US Economy», *Off. Assist. Secr. Plan. Eval. US Dep. Health Hum. Serv. Sept. Available Httpaspe Hhs Govhealthreports08healthcarecostreport Html HHS*, 2007.
- [20] D. Lupton, «The thing-power of the human-app health assemblage: thinking with vital materialism», *Soc. Theory Health*, vol. 17, n. 2, pp. 125–139, 2019.
- [21] K. Huckvale, C. J. Wang, A. Majeed, e J. Car, «Digital health at fifteen: more human (more needed)», *BMC Med.*, vol. 17, n. 1, p. 62, 2019.
- [22] N. Couldry e U. A. Mejias, «Data colonialism: Rethinking big data's relation to the contemporary subject», *Telev. New Media*, vol. 20, n. 4, pp. 336–349, 2019.
- [23] G. Hripcsak *et al.*, «Health data use, stewardship, and governance: ongoing gaps and challenges: a report from AMIA's 2012 Health Policy Meeting», *J. Am. Med. Inform. Assoc.*, vol. 21, n. 2, pp. 204–211, 2014.
- [24] L. R. Warren, J. M. Clarke, S. Arora, M. Barahona, N. Arebi, e A. Darzi, «Transitions of care across hospital settings in patients with inflammatory bowel disease», *World J. Gastroenterol.*, vol. 25, n. 17, p. 2122, 2019.
- [25] A. D. Beggs, E.-R. McGlone, e P. R. Thomas, «Impact of centralisation on vascular surgical services», *Br. J. Healthc. Manag.*, vol. 18, n. 9, pp. 468–473, 2012.
- [26] C. Vogt, M. Gersch, C. Spies, e K. Bengler, «Digital Transformation in Healthcare: How the Potential of Digital Health Is Tackled to Transform the Care Process of Intensive Care Patients Across All Healthcare Sectors», em *Digitalization Cases*, Springer, 2019, pp. 343–361.
- [27] G. L. Daumit, E. M. Stone, A. Kennedy-Hendricks, S. Choksy, J. A. Marsteller, e E. E. McGinty, «Care Coordination and Population Health Management Strategies and Challenges in a Behavioral Health Home Model», *Med. Care*, vol. 57, n. 1, pp. 79–84, 2019.
- [28] M. T. Mardini, Y. Iraqi, e N. Agoulmine, «A Survey of Healthcare Monitoring Systems for Chronically Ill Patients and Elderly», *J. Med. Syst.*, vol. 43, n. 3, p. 50, 2019.
- [29] P. Workman, A. A. Antolin, e B. Al-Lazikani, «Transforming cancer drug discovery with Big Data and AI», 2019.
- [30] H. Atasoy, B. N. Greenwood, e J. S. McCullough, «The digitization of patient care: A review of the effects of electronic health records on health care quality and utilization», *Annu. Rev. Public Health*, vol. 40, pp. 487–500, 2019.
- [31] V. M. M. Martins, «Integração de Sistemas de Informação»: p. 218.
- [32] M. Gold e C. McLAUGHLIN, «Assessing HITECH implementation and lessons: 5 years later», *Milbank Q.*, vol. 94, n. 3, pp. 654–687, 2016.
- [33] C. M. Angst e R. Agarwal, «Adoption of electronic health records in the presence of privacy concerns: The elaboration likelihood model and individual persuasion», *MIS Q.*, vol. 33, n. 2, pp. 339–370, 2009.
- [34] S. D. DeVore e K. Figlioli, «Lessons premier hospitals learned about implementing electronic health records», *Health Aff. (Millwood)*, vol. 29, n. 4, pp. 664–667, 2010.
- [35] Z. S. N. Reis, T. A. Maia, M. S. Marcolino, F. Becerra-Posada, D. Novillo-Ortiz, e A. L. P. Ribeiro, «Is There Evidence of Cost Benefits of Electronic Medical Records, Standards, or Interoperability in Hospital Information Systems? Overview of Systematic Reviews», *JMIR Med Inf.*, vol. 5, n. 3, p. e26, Ago. 2017.
- [36] M. S. Donaldson, J. M. Corrigan, e L. T. Kohn, *To err is human: building a safer health system*, vol. 6. National Academies Press, 2000.

- [37] W. Raghupathi e V. Raghupathi, «Big data analytics in healthcare: promise and potential», *Health Inf. Sci. Syst.*, vol. 2, n. 1, p. 3, Fev. 2014.
- [38] E. Shortliffe, *Computer-based medical consultations: MYCIN*, vol. 2. Elsevier, 2012.
- [39] E. H. Shortliffe e J. J. Cimino, *Biomedical informatics*. Springer, 2006.
- [40] E. H. Shortliffe e B. G. Buchanan, «A model of inexact reasoning in medicine», *Math. Biosci.*, vol. 23, n. 3-4, pp. 351-379, 1975.
- [41] R. Oliveira, D. Ferreira, R. Ferreira, e R. Cruz-Correia, «Open-source based integration solution for hospitals», apresentado na 2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS), 2016, pp. 294-299.
- [42] D. Lajas, D. N. Gonçalves-Ferreira, D. Ferreira, R. Oliveira, e R. Cruz-Correia, «Análise e Comparação de Mensagens HL7 em ambiente Hospitalar Analysis and Comparison of HL7 Messages in a Hospital Services».
- [43] D. O. C. Lajas, «Análise e comparação de mensagens HL7 em ambiente hospitalar», 2017.
- [44] J. D. Day e H. Zimmermann, «The OSI reference model», *Proc. IEEE*, vol. 71, n. 12, pp. 1334-1340, 1983.
- [45] F. D. C. Moreira, «Interoperabilidade em Sistemas de Informação na Saúde usando HL7», *Interoperability in health information systems using HL7*, 2014.
- [46] L.-F. Ko *et al.*, «HL7 middleware framework for healthcare information system», apresentado na HEALTHCOM 2006 8th International Conference on e-Health Networking, Applications and Services, 2006, pp. 152-156.
- [47] W.-Y. Yang, L.-H. Lee, H.-L. Gien, H.-Y. Chu, Y.-T. Chou, e D.-M. Liou, «The design of the hl7 rim-based sharing components for clinical information systems», *Int. J. Soc. Sci.*, vol. 5, n. 2, pp. 859-862, 2010.
- [48] «Health level seven international». [Em linha]. Disponível em: <http://www.hl7.org/about/index.cfm?ref=common>. [Acedido: 01-Abr-2019].
- [49] R. J. T. Ferreira, M. E. C. D. Correia, F. N. R. Gonçalves, e R. J. C. Correia, «Data Quality in HL7 Messages--A Real Case Analysis», apresentado na 2015 IEEE 28th International Symposium on Computer-Based Medical Systems, 2015, pp. 197-200.
- [50] K. Schurenberg, R. Yeager, e R. Johnson, «System and method for implementing a global master patient index», US20020007284A1, 17-Jan-2002.
- [51] C. Feied e F. Iskandar, «Master patient index», Nov. 2007.
- [52] B. Oleg e V. Kuttalingam, «Systems and methods for managing a master patient index including duplicate record detection», Set. 2015.
- [53] R. W. Maughan e J. B. Jacobsen, «Healthcare system and user interface for consolidating patient related information from different sources», Jul. 2010.
- [54] P. P. Ray, «Home Health Hub Internet of Things (H 3 IoT): an architectural framework for monitoring health of elderly people», apresentado na 2014 International Conference on Science Engineering and Management Research (ICSEMR), 2014, pp. 1-3.
- [55] M. Gupta, L. Chotard, Ó. Ingþórsson, J. Bastos, e I. Borges, «Health@Home - An e-Service Model for Disease Prevention and Healthcare in the Home», em *Electronic Healthcare*, 2009, pp. 17-24.
- [56] J. Street e R. Arcement, «Health Level Seven (HL7) Version 2.5.1 Guidelines», *La. Dep. Health Hosp. - Electron. Lab. Report. -ELR*, 2014.
- [57] T. Botsis, G. Hartvigsen, F. Chen, e C. Weng, «Secondary use of EHR: data quality issues and informatics opportunities», *Summit Transl. Bioinforma.*, vol. 2010, p. 1, 2010.
- [58] J. T. Finnell, J. M. Overhage, e S. Grannis, «All health care is not local: an evaluation of the distribution of Emergency Department care delivered in Indiana», apresentado na AMIA Annual Symposium Proceedings, 2011, vol. 2011, p. 409.

- [59] K. B. Wagholikar *et al.*, «Extending i2b2 into a framework for semantic abstraction of EHR to facilitate rapid development and portability of Health IT applications», *AMIA Jt. Summits Transl. Sci. Proc. AMIA Jt. Summits Transl. Sci.*, vol. 2019, pp. 370–378, Mai. 2019.
- [60] A. A. Donovan e B. W. Kernighan, *The Go programming language*. Addison-Wesley Professional, 2015.
- [61] R. J. Cruz-Correia, P. M. Vieira-Marques, A. M. Ferreira, F. C. Almeida, J. C. Wyatt, e A. M. Costa-Pereira, «Reviewing the integration of patient data: how systems are evolving in practice to meet patient needs», *BMC Med. Inform. Decis. Mak.*, vol. 7, n. 1, p. 14, Jun. 2007.
- [62] T. H. Payne, P. J. Hoey, P. Nichol, e C. Lovis, «Preparation and use of preconstructed orders, order sets, and order menus in a computerized provider order entry system», *J. Am. Med. Inform. Assoc.*, vol. 10, n. 4, pp. 322–329, 2003.
- [63] N. L. Downing *et al.*, «Electronic health record-based clinical decision support alert for severe sepsis: a randomised evaluation», *BMJ Qual Saf*, p. bmjqs-2018, 2019.
- [64] M. H. Hooper *et al.*, «Randomized trial of automated, electronic monitoring to facilitate early detection of sepsis in the intensive care unit», *Crit. Care Med.*, vol. 40, n. 7, p. 2096, 2012.
- [65] S. A. Sterling, W. R. Miller, J. Pryor, M. A. Puskarich, e A. E. Jones, «The impact of timing of antibiotics on outcomes in severe sepsis and septic shock: a systematic review and meta-analysis», *Crit. Care Med.*, vol. 43, n. 9, p. 1907, 2015.
- [66] J. C. Nelson *et al.*, «Challenges in the design and analysis of sequentially monitored postmarket safety surveillance evaluations using electronic observational health care data», *Pharmacoepidemiol. Drug Saf.*, vol. 21, pp. 62–71, 2012.
- [67] E. S. Kirkendall, Y. Ni, T. Lingren, M. Leonard, E. S. Hall, e K. Melton, «Data Challenges With Real-Time Safety Event Detection And Clinical Decision Support», *J. Med. Internet Res.*, vol. 21, n. 5, p. e13047, 2019.
- [68] H. J. Lowe, T. A. Ferris, P. M. Hernandez, e S. C. Weber, «STRIDE—An integrated standards-based translational research informatics platform», apresentado na AMIA Annual Symposium Proceedings, 2009, vol. 2009, p. 391.
- [69] J. D. Stein *et al.*, «Evaluation of an algorithm for identifying ocular conditions in electronic health record data», *JAMA Ophthalmol.*, vol. 137, n. 5, pp. 491–497, 2019.
- [70] D. Pinto, D. Vilarino, Y. Alemán, H. Gómez, e N. Loya, «The soundex phonetic algorithm revisited for sms-based information retrieval», apresentado na II Spanish Conference on Information Retrieval CERI, 2012.
- [71] A. J. Lait e B. Randell, «An assessment of name matching algorithms», *Tech. Rep. Ser.-Univ. Newctle. Tyne Comput. Sci.*, 1996.
- [72] Z. Bhatti, A. Waqas, I. A. Ismaili, D. N. Hakro, e W. J. Soomro, «Phonetic based soundex & shapeex algorithm for sindhi spell checker system», *ArXiv Prepr. ArXiv14053033*, 2014.
- [73] C. Chatwani, «Health care system for physicians to manage their patients», Mai. 2019.
- [74] R. Sindhwani, P. L. Singh, D. K. Prajapati, A. Iqbal, R. K. Phanden, e V. Malhotra, «Agile System in Health Care: Literature Review», em *Advances in Industrial and Production Engineering*, 2019, pp. 643–652.
- [75] G. de Freitas Nobre, M. Kalichsztein, e M. M. Ramos, «Hospital bed automation system and methods for performing signal read and correlation as well as real-time data processing», Mai. 2019.
- [76] C. Quantin *et al.*, «A peculiar aspect of patients' safety: the discriminating power of identifiers for record linkage», *Stud Health Technol Inf.*, vol. 103:400–6, 2004.
- [77] J. C. Wyatt, «Clinical data systems, Part 1: Data and medical records», *Lancet*, vol. (8936):154, 1994.

- [78] J. M. Ferranti, R. C. Musser, K. Kawamoto, e W. E. Hammond, «The clinical document architecture and the continuity of care record: a critical analysis», *J. Am. Med. Inform. Assoc.*, vol. 13, n. 3, pp. 245–252, 2006.
- [79] R. Cruz-Correia, P. Vieira-Marques, A. Ferreira, E. Oliveira-Palhares, P. Costa, e F. Araújo, «Automatic detection of patient data inconsistencies on integrated Health Information Systems», 2006.
- [80] R. Cruz-Correia, P. Vieira-Marques, e E. Oliveira-Palhares, «Monitoring the integration of hospital information systems: how it may ensure and improve the quality of data», *Med. Care N.a 3*, vol. 121, pp. 176–182, 2006.
- [81] P. R. Silva-Ferreira, P. M. Vieira-Marques, J. H. Patriarca-Almeida, e R. J. Cruz-Correia, «Improving expressiveness of agents using openEHR to retrieve multi-institutional health data: Feeding local repositories through HL7 based providers», apresentado na 7th Iberian Conference on Information Systems and Technologies (CISTI 2012), 2012, pp. 1–5.
- [82] P. M. Vieira-Marques, S. Robles, J. Cucurull, R. J. Cruz-Correia, G. Navarro, e R. Marti, «Secure integration of distributed medical data using mobile agents», *IEEE Intell. Syst.*, vol. 21, n. 6, pp. 47–54, 2006.
- [83] P. A. Maranhão, G. M. Bacelar-Silva, D. N. G. Ferreira, C. Calhau, P. Vieira-Marques, e R. J. Cruz-Correia, «Nutrigenomic information in the openEHR data set», *Appl. Clin. Inform.*, vol. 9, n. 01, pp. 221–231, 2018.
- [84] M.-F. Fortin e N. Salgueiro, *O processo de investigação: da concepção à realização*. 1999.
- [85] R. Oliveira, D. Ferreira, R. Ferreira, e R. Cruz-Correia, «Open-source based integration solution for hospitals», em *International Symposium on Computer-Based Medical Systems*, 2016.
- [86] HLTSYS, «HS.HELIOS». [Em linha]. Disponível em: <http://hltsys.pt/hs-helios-en/>. [Acedido: 22-Abr-2019].




```

        if (vt['VT_ACCENTS'] === 2 &&
binary_search(NAMES_PT, words[i], false) ) { // false: accents considered
            quality -= vt['VT_W1ST_NOACCENTS'];
            prob_description += "Known first name, but
without accents. ";
        } else {
            quality -= vt['VT_UNKW_FIRST'];
            prob_description += "Unknown first name. ";
        }
    }
    // First name length < VT_1ST_MIN_CHARS
    if (words[i].length !== 0 && words[i].length <
vt['VT_1ST_MIN_CHARS']) {
        quality -= vt['VT_1ST_LESS_C'];
        prob_description += "The first name has less then
" + vt['VT_1ST_MIN_CHARS'] + " character(s). ";
    }
}

    // Last name length < VT_LAST_MIN_CHARS
    for (i = words.length-1; i > 0 && words[i] !== ""; i--); //
Ignore spaces at the end
    var last_n = words[i];
    if (last_n.length !== 0 && last_n.length <
        vt['VT_LAST_MIN_CHARS']) {
        quality -= vt['VT_LAST_LESS_C'];
        prob_description += "The last name has less then " +
            vt['VT_LAST_MIN_CHARS'] + " character(s). ";
    }
    for (var i=0; i < words.length; i++) {
        if (words[i] !== "")
            count_words++;
        // Verify if word have only 1 consonant
        if ( words[i].length === 1 && words[i].match(/[a-z]/gi)
            && !words[i].match(/[aeiou]/gi) ) {
            quality -= vt['VT_1CONS'];
            prob_description += "Name with only one consonant.
";
        }
        // Verify if word is an invalid name
        if ( binary_search(INVALID_NAMES, words_no_accents[i],
false) ) {
            invalid_words ++;
        }
    }

    // Invalid name found
    if (invalid_words !== 0) {
        quality -= vt['VT_INVALID_NAME'];
        prob_description += "Invalid name. ";
    }

    // Number of words < VT_MIN_WORDS
    if (count_words < vt['VT_MIN_WORDS']) {
        quality -= vt['VT_LESS_W'];
        prob_description += "The name has less then " +
vt['VT_MIN_WORDS'] + " word(s). ";
    }

    for (var i=0; i < name_no_accents.length; i++) {
        // Verify invalid characters

```

```

        if (! name_no_accents[i].match(vt['VT_VALID_SCHARS'])) {
            count_inv_chars ++;
            invalid_chars += name[i];
        }
        else
        {
            // Verify hyphens in the beginning, the end, and
invalid characters before and after hiphen.
            if (name[i] == '-') {
                if (i==0) {
                    quality -= vt['VT_BEG_HIPHEN'];
                    prob_description += "The name begins
with a hyphen. ";
                }
                if (i==name.length-1) {
                    quality -= vt['VT_END_HIPHEN'];
                    prob_description += "The name ends with
a hyphen. ";
                }
                if ( i>0 && !name[i-1].match(/[a-z]/gi) ) {
                    quality -= vt['VT_IBEF_HIPHEN'];
                    prob_description += "Invalid character
before a hyphen. ";
                }
                if ( i<name.length-1 &&
!name[i+1].match(/[a-z]/gi) ) {
                    quality -= vt['VT_IAFT_HIPHEN'];
                    prob_description += "Invalid character
after a hyphen. ";
                }
            }
            // Verify spaces in the beginning and the end
            if (name[i] == ' ') {
                if (i==0) {
                    quality -= vt['VT_BEG_SPC'];
                    prob_description += "The name begins
with a space. ";
                }
                if (i==name.length-1) {
                    quality -= vt['VT_END_SPC'];
                    prob_description += "The name ends with
a space. ";
                }
            }
        }
    }
}

// Invalid characters found
var chars_tot = name.length-(words.length-1);
if (count_inv_chars > 0) {
    if (typeof vt['VT_INVALID_CHS'] === "string") {
        quality -= ( count_inv_chars / chars_tot );
    } else {
        quality -= vt['VT_INVALID_CHS'];
    }
    prob_description += count_inv_chars + " invalid
character(s) found: " + invalid_chars + ". ";
}

// Spaces together found
var spaces_together = name.match(/ /gi);

```

```

        if (spaces_together != null) {
            if (typeof vt['VT_SPC_TOGHT'] === "string") {
                quality -= ( (spaces_together.length*2) /
chars_tot );
            } else {
                quality -= vt['VT_SPC_TOGHT'];
            }
            prob_description += (spaces_together.length*2) + "
spaces together. ";
        }

        // Quality level never below 0%
        quality = (quality < 0 ? 0: quality);
        if (prob_description == "") {
            prob_description = "No errors detected. ";
        }
    }

    var endTime = Date.now();
    var timeGap = endTime - startTime;

    //var request      = new Object();
    tmp["patientName"]      = name;
    tmp["quality"]          = quality;
    tmp["description"]      = prob_description;
    tmp["timeGap"]          = timeGap;
    tmp["sendingApplication"] = msg["sendingApplication"].toString();
    tmp["receivingApplication"] =
msg["receivingApplication"].toString();
    tmp["messageID"]        = msg["messageID"].toString();
    //logger.info(JSON.stringify(request));
    //channelMap.put("request", JSON.stringify(request));

    return tmp;
}

function binary_search (array, toSearch, withAccents) {
    var middle = 0;
    var begin  = 0;
    var end    = array.length-1;

    while (begin <= end) {
        middle = Math.floor((begin + end)/2);
        var current = array[middle];
        if (!withAccents)
            current = current.normalize('NFD').replace(/[\u0300-
\u036f]/g, "");
        if (toSearch.localeCompare(current) == 0)
            return true;
        if (toSearch.localeCompare(current) < 0)
            end = middle - 1;
        else
            begin = middle + 1;
    }
    return false;
}

return validate_name(msg["patientName"]);

```

8.2. Algoritmo na Linguagem Java de Validação de Nomes de Utentes

8.2.1. NameQuality.java

```
package NameQuality;

import java.io.File;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class NameQuality {
    public static void main(String[] args) throws Exception{

        BaseFiles bf = BaseFiles.getInstance();
        Output of = Output.getInstance();
        Count count = Count.getInstance();

        String name;
        float quality;
        String probab_description;
        long timeGap;
        String sendingApplication;
        String receivingApplication;
        String messageId;
        String timeStamp;

        String words[];
        int index = 0;

        /* ****
        String namesPath = "/teste/";
        File readFx = new File(namesPath + "names.txt");
        Scanner readFile = new Scanner(readFx, "UTF-8");
        **** */

        // Loading names.csv file
        Path filePath = Paths.get(System.getProperty("user.dir"));
        filePath = Paths.get("/Program Files/Mirth Connect/");
        filePath = filePath.resolve("./projects/namequality/");
        String filePath = filePath.resolve("names.csv").normalize().toString();
```

```

File readFx = new File(filePath);
Scanner readFile = new Scanner(readFx, "UTF-8");
while (readFile.hasNextLine()) {
    name = readFile.nextLine();

    words = name.split(",");
    name = words[0];
    sendingApplication = "";
    receivingApplication = "";
    messageID = "";
    timeStamp = "";
    if (words.length-1 >= 1) { sendingApplication = words[1]; }
    if (words.length-1 >= 2) { receivingApplication = words[2]; }
    if (words.length-1 >= 3) { messageID = words[3]; }
    if (words.length-1 >= 4) { timeStamp = words[4]; }

    // *****
    CheckName msg = new CheckName(name, sendingApplication,
receivingApplication, messageID, timeStamp);
    // *****
    msg.validateName();
    index++;
}

readFile.close();

if (bf.OUTPUT_REPORT || bf.ALL_OUTPUT) {
    SimpleDateFormat formatter= new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
    Date date = new Date();
    of.writeFile.write("Name Quality Report: " + formatter.format(date));
    of.writeFile.write("\n\nTotal names: " + index + "\n");

    of.writeFile.write("Is OK: " + count.N_NO_ERRORS + "\n");
    of.writeFile.write("Has Invalid Names: " + count.N_INVALID_NAME + "\n");
    of.writeFile.write("Has invalid Caracteres: " + count.N_INVALID_CHS + "\n");
    of.writeFile.write("Is Unknown Patients: " + count.N_UNKNOWN_PATIENT +
"\n");
    of.writeFile.write("Is NewBorn: " + count.N_NEWBORN + "\n");
    of.writeFile.write("Is Test: " + count.N_IS_TEST + "\n");
    of.writeFile.write("Is Maked: " + count.N_IS_MARKED + "\n\n");

    of.writeFile.write("The name field is empty.: " + count.N_EMPTY + "\n");
    of.writeFile.write("Known first name, but without accents.: " +
count.N_W1ST_NOACCENTS + "\n");
    of.writeFile.write("Unknown first name.: " + count.N_UNKW_FIRST + "\n");
    of.writeFile.write("The first name has less then " + bf.VT_1ST_MIN_CHARS + "
character(s).: " + count.N_1ST_LESS_C + "\n");
    of.writeFile.write("The last name has less then " + bf.VT_LAST_MIN_CHARS + "
characters.: " + count.N_LAST_LESS_C + "\n");

```


8.2.2. CheckName.java

```
package NameQuality;

import java.text.Normalizer;

public class CheckName {
    String name;
    float quality;
    String prob_description;
    long timeGap;
    String sendingApplication;
    String receivingApplication;
    String messageID;
    String timeStamp;

    public CheckName(String name, float quality, String prob_description, long timeGap,
String sendingApplication, String receivingApplication, String messageID, String
timeStamp) {
        this.name = name;
        this.quality = quality;
        this.prob_description = prob_description;
        this.timeGap = timeGap;
        this.sendingApplication = sendingApplication;
        this.receivingApplication = receivingApplication;
        this.messageID = messageID;
        this.timeStamp = timeStamp;
    }

    public CheckName(String name, String sendingApplication, String
receivingApplication, String messageID, String timeStamp) {
        this.name = name;
        this.quality = 1;
        this.prob_description = "";
        this.timeGap = 0;
        this.sendingApplication = sendingApplication;
        this.receivingApplication = receivingApplication;
        this.messageID = messageID;
        this.timeStamp = timeStamp;
    }

    public CheckName(String name) {
        this.name = name;
        this.quality = 1;
        this.prob_description = "";
        this.timeGap = 0;
        this.sendingApplication = "";
        this.receivingApplication = "";
        this.messageID = "";
        this.timeStamp = "";
    }
}
```



```
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public float getQuality() {
    return quality;
}

public void setQuality(float quality) {
    this.quality = quality;
}

public String getProb_description() {
    return prob_description;
}

public void setProb_description(String prob_description) {
    this.prob_description = prob_description;
}

public long getTimeGap() {
    return timeGap;
}

public void setTimeGap(long timeGap) {
    this.timeGap = timeGap;
}

public String getSendingApplication() {
    return sendingApplication;
}

public void setSendingApplication(String sendingApplication) {
    this.sendingApplication = sendingApplication;
}

public String getReceivingApplication() {
    return receivingApplication;
}

public void setReceivingApplication(String receivingApplication) {
    this.receivingApplication = receivingApplication;
}

public String getMessageID() {
```

```

        return messageID;
    }

    public void setMessageID(String messageID) {
        this.messageID = messageID;
    }

    public String getTimeStamp() {
        return timeStamp;
    }

    public void setTimeStamp(String timeStamp) {
        this.timeStamp = timeStamp;
    }

    public void validateName() throws Exception {
        long    startTime = System.currentTimeMillis();

        BaseFiles bf = BaseFiles.getInstance();
        //System.out.println("Name: " + this.name + " bf.VT_ACCENTS: " + bf.VT_ACCENTS);

        Output    of = Output.getInstance();
        Count    count = Count.getInstance();

        int    aux = 0; // output words with invalid characters
        long    endTime;
        String    nameOut;
        String    words[];

        int    count_inv_chars = 0;
        String    invalid_chars = "";
        int    count_words = 0;
        int    invalid_words = 0;
        boolean    accents = (bf.VT_ACCENTS == 0 ? false : true);
        int    item;
        String    name_no_accents;
        String[]    words_no_accents;
        int    spaces_together = 0;
        String    last_n;
        int    search;
        String    letter;
        boolean    spaces = false;

        if (this.name.equals("")) {
            this.quality -= bf.VT_EMPTY;
            this.prob_description += "The name field is empty. ";
            count.N_EMPTY++;
        }
        else
        { // Remove accents

```



```

        of.f_04fn_lc.write(name + "\n");
    }
}

// Last name length < VT_LAST_MIN_CHARS
for (item = words.length-1; item > 0 && words[item].equals(""); ) item--; //
Ignore spaces at the end
last_n = words[item];
if (last_n.length() != 0 && last_n.length() < bf.VT_LAST_MIN_CHARS) {
    this.quality -= bf.VT_LAST_LESS_C;
    this.prob_description += "The last name has less then " +
bf.VT_LAST_MIN_CHARS + " character(s). ";
    count.N_LAST_LESS_C++;
    if (bf.ALL_OUTPUT) {
        of.f_05ln_lc.write(name + "\n");
    }
}
for (item=0; item < words.length; item++) {
    if ( !words[item].equals("") )
        count_words++;
    // Verify if word have only 1 consonant
    if ( words[item].length() == 1 && words[item].matches("[a-z]") &&
!words[item].matches("[aeiou]") ) {
        this.quality -= bf.VT_1CONS;
        this.prob_description += "Name with only one consonant. ";
        count.N_1CONS++;
        if (bf.ALL_OUTPUT) {
            of.f_06_1cons.write(name + "\n");
        }
    }
    // Verify if word is an invalid name
    if ( bf.INVALID_NAMES.indexOf(words_no_accents[item]) > -1 ) {
        invalid_words ++;
    }
}

// Invalid name found
if (invalid_words != 0) {
    this.quality -= bf.VT_INVALID_NAME;
    this.prob_description += "Invalid name. ";
    count.N_INVALID_NAME++;
    if (bf.ALL_OUTPUT) {
        of.f_07inv_n.write(name + "\n");
    }
}

// Number of words < VT_MIN_WORDS
if (count_words < bf.VT_MIN_WORDS) {
    this.quality -= bf.VT_LESS_W;

```

```

        this.prob_description += "The name has less then " + bf.VT_MIN_WORDS + "
word(s). ";
        count.N_LESS_W++;
        if (bf.ALL_OUTPUT) {
            of.f_08lwords.write(name + "\n");
        }
    }

    for (item=0; item < name_no_accents.length(); item++) {
        // Verify invalid characters
        letter = Character.toString(name_no_accents.charAt(item));
        if ( !letter.equals(" ") && !letter.matches(bf.VT_VALID_SCHARS) &&
!letter.matches("[a-z]") ) {
            count_inv_chars ++;
            invalid_chars += this.name.charAt(item);
            /*for (aux = 0; aux < words.length; aux++) {
            if (words[aux].indexOf(name.charAt(i)) > 0) {
                writeFile.write(words[aux] + "\n");
            }
            } */
        }
        else
        {
            // Verify hyphens in the beginning, the end, and invalid characters before
and after hiphen.
            if (this.name.charAt(item) == '-') {
                if (item == 0) {
                    this.quality -= bf.VT_BEG_HIPHEN;
                    this.prob_description += "The name begins with a hyphen. ";
                    count.N_BEG_HIPHEN++;
                    if (bf.ALL_OUTPUT) {
                        of.f_09b_hif.write(name + "\n");
                    }
                }
                if (item == this.name.length()-1) {
                    this.quality -= bf.VT_END_HIPHEN;
                    this.prob_description += "The name ends with a hyphen. ";
                    count.N_END_HIPHEN++;
                    if (bf.ALL_OUTPUT) {
                        of.f_10e_hif.write(name + "\n");
                    }
                }
            }
            if ( item>0 && !Character.toString(this.name.charAt(item-1)).matches("/[a-
z]/gi") ) {

                this.quality -= bf.VT_IBEF_HIPHEN;
                this.prob_description += "Invalid character before a hyphen. ";
                count.N_IBEF_HIPHEN++;
                if (bf.ALL_OUTPUT) {
                    of.f_11inv_bhif.write(name + "\n");
                }
            }
        }
    }

```

```

        if ( item<this.name.length()-1 &&
!Character.toString(this.name.charAt(item+1)).matches("/[a-z]/gi") ) {
            this.quality -= bf.VT_IAFT_HIPHEN;
            this.prob_description += "Invalid character after a hyphen. ";
            count.N_IAFT_HIPHEN++;
            if (bf.ALL_OUTPUT) {
                of.f_12inv_ahif.write(name + "\n");
            }
        }
    }
    // Verify spaces in the beginning and the end
    if (spaces && this.name.charAt(item) == ' ') {
        if (item == 0) {
            this.quality -= bf.VT_BEG_SPC;
            this.prob_description += "The name begins with a space. ";
            count.N_BEG_SPC++;
            if (bf.ALL_OUTPUT) {
                of.f_13b_spc.write(name + "\n");
            }
        }
        if (item == this.name.length()-1) {
            this.quality -= bf.VT_END_SPC;
            this.prob_description += "The name ends with a space. ";
            count.N_END_SPC++;
            if (bf.ALL_OUTPUT) {
                of.f_14e_spc.write(name + "\n");
            }
        }
        if (item+1 < this.name.length())
            if (this.name.charAt(item+1) == ' ') {
                spaces_together++;
            }
    }
}

// Invalid characters found
int chars_tot = this.name.length()-(words.length-1);
if (count_inv_chars > 0) {
    if (bf.VT_INVALID_CHS.equals("C")) {
        this.quality -= ( count_inv_chars / chars_tot );
    } else {
        this.quality -= Float.parseFloat(bf.VT_INVALID_CHS);
    }
    this.prob_description += count_inv_chars + " invalid character(s) found: " +
invalid_chars + ". ";
    count.N_INVALID_CHS++;
    if (bf.ALL_OUTPUT) {
        of.f_15inv_c.write(name + "\n");
    }
}

```

```

// Spaces together found
if (spaces && spaces_together != 0) {
    if (bf.VT_SPC_TOGHT.equals("C")) {
        this.quality -= (spaces_together*2) / (float)chars_tot);
    } else {
        this.quality -= Float.parseFloat(bf.VT_SPC_TOGHT);
    }
    this.prob_description += (spaces_together*2) + " spaces together. ";
    count.N_SPC_TOGHT++;
    if (bf.ALL_OUTPUT) {
        of.f_16spc_tg.write(name + "\n");
    }
}

// Quality level never below 0%
this.quality = (this.quality < 0 ? 0: this.quality);
if (this.prob_description.equals("")) {
    this.prob_description = "No errors detected. ";
    count.N_NO_ERRORS++;
    if (bf.ALL_OUTPUT) {
        of.f_17no_errors.write(name + "\n");
    }
}
}

endTime = System.currentTimeMillis();
this.timeGap = endTime-startTime;

if (bf.ALL_OUTPUT) {
    nameOut = this.name + "," + this.quality + "," + this.prob_description + "," +
this.timeGap + "," + this.sendingApplication + "," + this.receivingApplication + "," +
this.messageID + "," + this.timeStamp;
    of.writeFile.write(nameOut + "\n");
}
/*
if (isOk())
    count.N_OK++;

if (hasInvalidName())
    count.N_INVALID_NAMES++;

if (hasInvalidCharacters())
    count.N_INVALID_CHARS++;
*/

if (this.unknownPatient())
    count.N_UNKNOWN_PATIENT++;

if (this.newBorn())
    count.N_NEWBORN++;

```

```

        if (this.isTest())
            count.N_IS_TEST++;

        if (this.isMarked())
            count.N_IS_MARKED++;
    }

    public boolean isOk() {
        try {
            this.validateName();
        } catch (Exception e) {};
        return (this.quality == 1);
    }

    public boolean hasInvalidName() {
        BaseFiles bf = BaseFiles.getInstance();

        int item;
        String name_no_accents = Normalizer.normalize(this.name.toLowerCase(),
Normalizer.Form.NFD);
        name_no_accents = name_no_accents.replaceAll("\\p{M}", "");
        String[] words_no_accents = name_no_accents.split(" ");

        for (item=0; item < words_no_accents.length; item++) {
            if ( bf.INVALID_NAMES.indexOf(words_no_accents[item]) > -1 ) {
                return (true);
            }
        }
        return (false);
    }

    public boolean hasInvalidCharacters() {
        BaseFiles bf = BaseFiles.getInstance();

        int item;
        String letter;
        String name_no_accents = Normalizer.normalize(this.name.toLowerCase(),
Normalizer.Form.NFD);
        name_no_accents = name_no_accents.replaceAll("\\p{M}", "");

        for (item=0; item < name_no_accents.length(); item++) {
            letter = Character.toString(name_no_accents.charAt(item));
            if ( !letter.equals(" ") && !letter.matches(bf.VT_VALID_SCHARS) &&
!letter.matches("[a-z]") ) {
                return (true);
            }
        }
        return (false);
    }
}

```



```

    public boolean unknownPatient() {
        String name_no_accents = Normalizer.normalize(this.name.toLowerCase(),
Normalizer.Form.NFD);
        name_no_accents = name_no_accents.replaceAll("\\p{M}", "");
        String[] words = name_no_accents.split(" ");
        boolean check = name_no_accents.equals("");
        check = check || name_no_accents.contains("nome utente");
        check = check || name_no_accents.contains("sem identificacao");
        check = check || name_no_accents.contains("nao identificado");
        check = check || name_no_accents.contains("fem nao identificado");
        check = check || name_no_accents.contains("masc nao identificado");
        check = check || name_no_accents.contains("fem identificado nao");
        check = check || name_no_accents.contains("masc identificado nao");
        return (check);
    }

    public boolean newBorn() {
        String name_aux = this.name.toLowerCase();
        boolean check = name_aux.startsWith("filho ");
        check = check || name_aux.startsWith("filha ");
        check = check || name_aux.startsWith("filho de ");
        check = check || name_aux.startsWith("filha de ");
        return (check);
    }

    public boolean isTest() {
        String name_no_accents = Normalizer.normalize(this.name.toLowerCase(),
Normalizer.Form.NFD);
        name_no_accents = name_no_accents.replaceAll("\\p{M}", "");
        String[] words = name_no_accents.split(" ");
        boolean check = false;
        for (String word : words) {
            check = check || word.equals("teste");
            check = check || word.equals("ficticio");
            check = check || word.equals("informatica");
            check = check || word.equals("integracao");
            check = check || word.equals("funcionario");
            check = check || word.equals("carro");
            check = check || word.equals("urgencia");
            check = check || word.equals("ucip");
            check = check || word.equals("controle");
            check = check || word.equals("bolsa");
            check = check || word.equals("nutricao");
            check = check || word.equals("parenterica");
            check = check || word.equals("rastreo");
            check = check || word.equals("oftalmologia");
            check = check || word.equals("uk neqas");
            // check = check || word.equals("neqas");
            check = check || word.equals("cardiaco");
            check = check || word.equals("bioquimica");
            check = check || word.equals("geral");
        }
    }

```

```
        check = check || word.equals("hemoglobina");
        check = check || word.equals("glicada");
        check = check || word.equals("qualidade");
        check = check || word.equals("imuno");
        check = check || word.equals("hla-b");
        check = check || word.equals("gasimetria");
    }
    return (check);
}

public boolean isMarked() {
    String name_no_accents = Normalizer.normalize(this.name.toLowerCase(),
Normalizer.Form.NFD);
    name_no_accents = name_no_accents.replaceAll("\\p{M}", "");
    String[] words = name_no_accents.split(" ");

    boolean check = name_no_accents.startsWith("seg ");
    check = check || name_no_accents.equals("seg");
    check = check || name_no_accents.equals("prim");
    for (String word : words) {
        check = check || word.equals("rnf");
        check = check || word.equals("rnm");
        check = check || word.equals("seg rnf");
        check = check || word.equals("seg rnm");
        check = check || word.equals("prim rnf");
        check = check || word.equals("prim rnm");
    }
    return (check);
}
}
```

8.2.3. BaseFiles.java

```

package NameQuality;

import java.io.File;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.text.Normalizer;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.Scanner;

public class BaseFiles {
    private static BaseFiles instance;

    List <String> NAMES_PT = new ArrayList();
    List <String> NAMES_PT_NO_ACCENTS = new ArrayList();
    List <String> INVALID_NAMES = new ArrayList();

    int VT_ACCENTS; // Accentuation considered (0=No or 1=Yes or 2=Both).
    double VT_W1ST_NOACCENTS; // Weight to recognize first name, but
without accents.
    int VT_MIN_WORDS; // Minimum number of words in the name.
    double VT_LESS_W; // The name has less then _VT_MIN_WORDS+1 words.
    int VT_1ST_MIN_CHARS; // Minimum number of characters in the first
name.
    double VT_1ST_LESS_C; // The first name has less then
_VT_1ST_MIN_CHARS characters.
    int VT_LAST_MIN_CHARS; // Minimum number of characters in the last
name.
    double VT_LAST_LESS_C; // The last name has less then
_VT_LAST_MIN_CHARS characters.
    double VT_EMPTY; // The name field is empty.
    double VT_UNKW_FIRST; // Unknown first name.
    double VT_1CONS; // Name with only one consonant.
    double VT_INVALID_NAME; // Invalid name.
    double VT_BEG_HIPHEN; // The name begins with a hyphen.
    double VT_END_HIPHEN; // The name ends with a hyphen.
    double VT_IBEF_HIPHEN; // Invalid character before a hyphen.
    double VT_Iaft_HIPHEN; // Invalid character after a hyphen.
    double VT_BEG_SPC; // The name begins with a space.
    double VT_END_SPC; // The name ends with a space.
    String VT_INVALID_CHS; // [N] invalid characters found. // N;
C=(nr_invalid_chars/total_chars); W=(nr_words_with_invalid_chars/total_words)
    String VT_SPC_TOGHT; // [N] spaces together. // N;
C=(nr_invalid_chars/total_chars); W=(nr_words_with_invalid_chars/total_words)
    String VT_VALID_SCHARS; // Valid special characters

```

```

boolean ALL_OUTPUT;
boolean OUTPUT_REPORT;

BaseFiles() throws Exception {
    this.VT_ACCENTS      = 2;          // Accentuation considered (0=No or 1=Yes or
2=Both).
    this.VT_W1ST_NOACCENTS = 0.2;      // Weight to recognize first
name, but without accents.
    this.VT_MIN_WORDS     = 2;          // Minimum number of words in the
name.
    this.VT_LESS_W        = 0.65;      // The name has less than
_VT_MIN_WORDS+1 words.
    this.VT_1ST_MIN_CHARS = 3;          // Minimum number of characters in the
first name.
    this.VT_1ST_LESS_C    = 0.35;      // The first name has less than
_VT_1ST_MIN_CHARS characters.
    this.VT_LAST_MIN_CHARS = 2;          // Minimum number of characters in the
last name.
    this.VT_LAST_LESS_C   = 0.55;      // The last name has less than
_VT_LAST_MIN_CHARS characters.
    this.VT_EMPTY         = 1;          // The name field is empty.
    this.VT_UNKW_FIRST     = 0.6;        // Unknown first name.
    this.VT_1CONS          = 0.7;        // Name with only one consonant.
    this.VT_INVALID_NAME   = 1;          // Invalid name.
    this.VT_BEG_HIPHEN     = 0.40;      // The name begins with a hyphen.
    this.VT_END_HIPHEN     = 0.42;      // The name ends with a hyphen.
    this.VT_IBEF_HIPHEN    = 0.44;      // Invalid character before a hyphen.
    this.VT_Iaft_HIPHEN    = 0.46;      // Invalid character after a hyphen.
    this.VT_BEG_SPC        = 0.1;        // The name begins with a space.
    this.VT_END_SPC        = 0.12;      // The name ends with a space.
    this.VT_INVALID_CHS    = "C";        // [N] invalid characters found. // N;
C=(nr_invalid_chars/total_chars); W=(nr_words_with_invalid_chars/total_words)
    this.VT_SPC_TOGHT      = "C";        // [N] spaces together. //
N; C=(nr_invalid_chars/total_chars); W=(nr_words_with_invalid_chars/total_words)
    this.VT_VALID_SCHARS   = "-\\\""; // Valid special characters

    this.ALL_OUTPUT        = false;
    this.OUTPUT_REPORT     = true;

    String line;
    String[] words;

    // Loading validation table file
    Path filePath = Paths.get(System.getProperty("user.dir"));
    filePath = Paths.get("/Program Files/Mirth Connect/");
    filePath = filePath.resolve("../projects/namequality/");
    String file_path = filePath.resolve("ValidationTable.txt").normalize().toString();

    File readFx = new File(file_path);

```

```

Scanner readFile = new Scanner(readFx, "UTF-8");
while ( readFile.hasNextLine() ){
    line = readFile.nextLine();
    words = line.split(";");
    switch(words[1]) {
        case "VT_ACCENTS": this.VT_ACCENTS = Integer.parseInt(words[0]); break;
        case "VT_W1ST_NOACCENTS": this.VT_W1ST_NOACCENTS =
Double.parseDouble(words[0]); break;
        case "VT_MIN_WORDS": this.VT_MIN_WORDS = Integer.parseInt(words[0]);
break;
        case "VT_LESS_W": this.VT_LESS_W = Double.parseDouble(words[0]); break;
        case "VT_1ST_MIN_CHARS": this.VT_1ST_MIN_CHARS =
Integer.parseInt(words[0]); break;
        case "VT_1ST_LESS_C": this.VT_1ST_LESS_C = Double.parseDouble(words[0]);
break;
        case "VT_LAST_MIN_CHARS": this.VT_LAST_MIN_CHARS =
Integer.parseInt(words[0]); break;
        case "VT_LAST_LESS_C": this.VT_LAST_LESS_C =
Double.parseDouble(words[0]); break;
        case "VT_EMPTY": this.VT_EMPTY = Double.parseDouble(words[0]); break;
        case "VT_UNKW_FIRST": this.VT_UNKW_FIRST =
Double.parseDouble(words[0]); break;
        case "VT_1CONS": this.VT_1CONS = Double.parseDouble(words[0]); break;
        case "VT_INVALID_NAME": this.VT_INVALID_NAME =
Double.parseDouble(words[0]); break;
        case "VT_BEG_HIPHEN": this.VT_BEG_HIPHEN =
Double.parseDouble(words[0]); break;
        case "VT_END_HIPHEN": this.VT_END_HIPHEN =
Double.parseDouble(words[0]); break;
        case "VT_IBEF_HIPHEN": this.VT_IBEF_HIPHEN =
Double.parseDouble(words[0]); break;
        case "VT_IAFT_HIPHEN": this.VT_IAFT_HIPHEN =
Double.parseDouble(words[0]); break;
        case "VT_BEG_SPC": this.VT_BEG_SPC = Double.parseDouble(words[0]);
break;
        case "VT_END_SPC": this.VT_END_SPC = Double.parseDouble(words[0]);
break;
        case "VT_INVALID_CHS": this.VT_INVALID_CHS = words[0]; break;
        case "VT_SPC_TOGHT": this.VT_SPC_TOGHT = words[0]; break;
        case "VT_VALID_SCHARS": this.VT_VALID_SCHARS = words[0]; break;
    }
}
readFile.close();

// Loading base files
this.NAMES_PT = new ArrayList();
filePath = filePath.resolve("names_pt.txt").normalize().toString();
readFx = new File(filePath);
readFile = new Scanner(readFx, "UTF-8");
this.NAMES_PT_NO_ACCENTS = new ArrayList();
while ( readFile.hasNextLine() ){

```

```

        line = readFile.nextLine().toLowerCase();
        if (this.VT_ACCENTS == 0) { // Ignore accents and remove them
            line = Normalizer.normalize(line, Normalizer.Form.NFD);
            line = line.replaceAll("\\p{M}", "");
        }
        this.NAMES_PT.add(line);
        if (this.VT_ACCENTS == 0 || this.VT_ACCENTS == 2) {
            line = Normalizer.normalize(line, Normalizer.Form.NFD);
            line = line.replaceAll("\\p{M}", "");
            this.NAMES_PT_NO_ACCENTS.add(line);
        }
    }
    readFile.close();

    this.INVALID_NAMES = new ArrayList();
    filePath = filePath.resolve("invalid_names.txt").normalize().toString();
    readFx = new File(filePath);
    readFile = new Scanner(readFx, "UTF-8");
    while (readFile.hasNextLine()) {
        line = readFile.nextLine().toLowerCase();
        line = Normalizer.normalize(line, Normalizer.Form.NFD);
        line = line.replaceAll("\\p{M}", "");
        this.INVALID_NAMES.add(line);
    }
    readFile.close();
}

static {
    try {
        instance = new BaseFiles();
    } catch (Exception e) {}
}

/** Static 'instance' method */
public static BaseFiles getInstance() {
    return instance;
}

public List <String> getNAMES_PT() {
    return NAMES_PT;
}

public void setNAMES_PT(List NAMES_PT) {
    this.NAMES_PT = NAMES_PT;
}

public List getNAMES_PT_NO_ACCENTS() {
    return NAMES_PT_NO_ACCENTS;
}

public void setNAMES_PT_NO_ACCENTS(List NAMES_PT_NO_ACCENTS) {

```

```
    this.NAMES_PT_NO_ACCENTS = NAMES_PT_NO_ACCENTS;
}

public List getINVALID_NAMES() {
    return INVALID_NAMES;
}

public void setINVALID_NAMES(List INVALID_NAMES) {
    this.INVALID_NAMES = INVALID_NAMES;
}

public int getVT_ACCENTS() {
    return VT_ACCENTS;
}

public void setVT_ACCENTS(int VT_ACCENTS) {
    this.VT_ACCENTS = VT_ACCENTS;
}

public double getVT_W1ST_NOACCENTS() {
    return VT_W1ST_NOACCENTS;
}

public void setVT_W1ST_NOACCENTS(double VT_W1ST_NOACCENTS) {
    this.VT_W1ST_NOACCENTS = VT_W1ST_NOACCENTS;
}

public int getVT_MIN_WORDS() {
    return VT_MIN_WORDS;
}

public void setVT_MIN_WORDS(int VT_MIN_WORDS) {
    this.VT_MIN_WORDS = VT_MIN_WORDS;
}

public double getVT_LESS_W() {
    return VT_LESS_W;
}

public void setVT_LESS_W(double VT_LESS_W) {
    this.VT_LESS_W = VT_LESS_W;
}

public int getVT_1ST_MIN_CHARS() {
    return VT_1ST_MIN_CHARS;
}

public void setVT_1ST_MIN_CHARS(int VT_1ST_MIN_CHARS) {
    this.VT_1ST_MIN_CHARS = VT_1ST_MIN_CHARS;
}
```

```
public double getVT_1ST_LESS_C() {
    return VT_1ST_LESS_C;
}

public void setVT_1ST_LESS_C(double VT_1ST_LESS_C) {
    this.VT_1ST_LESS_C = VT_1ST_LESS_C;
}

public int getVT_LAST_MIN_CHARS() {
    return VT_LAST_MIN_CHARS;
}

public void setVT_LAST_MIN_CHARS(int VT_LAST_MIN_CHARS) {
    this.VT_LAST_MIN_CHARS = VT_LAST_MIN_CHARS;
}

public double getVT_LAST_LESS_C() {
    return VT_LAST_LESS_C;
}

public void setVT_LAST_LESS_C(double VT_LAST_LESS_C) {
    this.VT_LAST_LESS_C = VT_LAST_LESS_C;
}

public double getVT_EMPTY() {
    return VT_EMPTY;
}

public void setVT_EMPTY(int VT_EMPTY) {
    this.VT_EMPTY = VT_EMPTY;
}

public double getVT_UNKW_FIRST() {
    return VT_UNKW_FIRST;
}

public void setVT_UNKW_FIRST(double VT_UNKW_FIRST) {
    this.VT_UNKW_FIRST = VT_UNKW_FIRST;
}

public double getVT_1CONS() {
    return VT_1CONS;
}

public void setVT_1CONS(double VT_1CONS) {
    this.VT_1CONS = VT_1CONS;
}

public double getVT_INVALID_NAME() {
    return VT_INVALID_NAME;
}
```



```
public void setVT_INVALID_NAME(int VT_INVALID_NAME) {
    this.VT_INVALID_NAME = VT_INVALID_NAME;
}

public double getVT_BEG_HIPHEN() {
    return VT_BEG_HIPHEN;
}

public void setVT_BEG_HIPHEN(double VT_BEG_HIPHEN) {
    this.VT_BEG_HIPHEN = VT_BEG_HIPHEN;
}

public double getVT_END_HIPHEN() {
    return VT_END_HIPHEN;
}

public void setVT_END_HIPHEN(double VT_END_HIPHEN) {
    this.VT_END_HIPHEN = VT_END_HIPHEN;
}

public double getVT_IBEF_HIPHEN() {
    return VT_IBEF_HIPHEN;
}

public void setVT_IBEF_HIPHEN(double VT_IBEF_HIPHEN) {
    this.VT_IBEF_HIPHEN = VT_IBEF_HIPHEN;
}

public double getVT_IAFT_HIPHEN() {
    return VT_IAFT_HIPHEN;
}

public void setVT_IAFT_HIPHEN(double VT_IAFT_HIPHEN) {
    this.VT_IAFT_HIPHEN = VT_IAFT_HIPHEN;
}

public double getVT_BEG_SPC() {
    return VT_BEG_SPC;
}

public void setVT_BEG_SPC(double VT_BEG_SPC) {
    this.VT_BEG_SPC = VT_BEG_SPC;
}

public double getVT_END_SPC() {
    return VT_END_SPC;
}

public void setVT_END_SPC(double VT_END_SPC) {
    this.VT_END_SPC = VT_END_SPC;
}
```

```
    }

    public String getVT_INVALID_CHS() {
        return VT_INVALID_CHS;
    }

    public void setVT_INVALID_CHS(String VT_INVALID_CHS) {
        this.VT_INVALID_CHS = VT_INVALID_CHS;
    }

    public String getVT_SPC_TOGHT() {
        return VT_SPC_TOGHT;
    }

    public void setVT_SPC_TOGHT(String VT_SPC_TOGHT) {
        this.VT_SPC_TOGHT = VT_SPC_TOGHT;
    }

    public String getVT_VALID_SCHARS() {
        return VT_VALID_SCHARS;
    }

    public void setVT_VALID_SCHARS(String VT_VALID_SCHARS) {
        this.VT_VALID_SCHARS = VT_VALID_SCHARS;
    }

    public void listNamesPT() {
        Iterator <String> iterator = this.NAMES_PT.iterator();
        while(iterator.hasNext()) {
            System.out.println(iterator.next());
        }
    }
}
```

8.2.4. Count.java

```

package NameQuality;

public class Count {
    private static Count instance;

    long N_NO_ERRORS;    // No errors detected.        / A - is Ok?
    long N_UNKNOWN_PATIENT; // D - is unknown?
    long N_NEWBORN;      // E - is a newBorn?
    long N_IS_TEST;      // F - is a Test?
    long N_IS_MARKED;    // G - is Marked?

    long N_EMPTY;        // The name field is empty.
    long N_W1ST_NOACCENTS; // Known first name, but without accents.
    long N_UNKW_FIRST;    // Unknown first name.
    long N_1ST_LESS_C;    // The first name has less then [N] character(s).
    long N_LAST_LESS_C;   // The last name has less then [N] characters.
    long N_1CONS;         // Name with only one consonant.
    long N_INVALID_NAME;  // Invalid name.                / B - has Invalid Name?
    long N_LESS_W;        // The name has less then [N] words.
    long N_BEG_HIPHEN;    // The name begins with a hyphen.
    long N_END_HIPHEN;    // The name ends with a hyphen.
    long N_IBEF_HIPHEN;   // Invalid character before a hyphen.
    long N_Iaft_HIPHEN;   // Invalid character after a hyphen.
    long N_BEG_SPC;       // The name begins with a space.
    long N_END_SPC;       // The name ends with a space.
    long N_INVALID_CHS;   // [N] invalid characters found.    / C - has Invalid
Characters?
    long N_SPC_TOGHT;     // [N] spaces together.

    private Count() {
        N_NO_ERRORS      = 0;
        N_UNKNOWN_PATIENT = 0;
        N_NEWBORN         = 0;
        N_IS_TEST         = 0;
        N_IS_MARKED       = 0;

        N_EMPTY          = 0;
        N_W1ST_NOACCENTS = 0;
        N_UNKW_FIRST     = 0;
        N_1ST_LESS_C     = 0;
        N_LAST_LESS_C    = 0;
        N_1CONS          = 0;
        N_INVALID_NAME   = 0;
        N_LESS_W         = 0;
        N_BEG_HIPHEN     = 0;
        N_END_HIPHEN     = 0;
        N_IBEF_HIPHEN    = 0;
        N_Iaft_HIPHEN    = 0;
    }
}

```

```
N_BEG_SPC      = 0;
N_END_SPC      = 0;
N_INVALID_CHS  = 0;
N_SPC_TOGHT    = 0;
}

static {
    instance = new Count();
}

/** Static 'instance' method */
public static Count getInstance() {
    return instance;
}

public long getN_NO_ERRORS() {
    return N_NO_ERRORS;
}

public void setN_NO_ERRORS(long N_NO_ERRORS) {
    this.N_NO_ERRORS = N_NO_ERRORS;
}

public long getN_UNKNOWN_PATIENT() {
    return N_UNKNOWN_PATIENT;
}

public void setN_UNKNOWN_PATIENT(long N_UNKNOWN_PATIENT) {
    this.N_UNKNOWN_PATIENT = N_UNKNOWN_PATIENT;
}

public long getN_NEWBORN() {
    return N_NEWBORN;
}

public void setN_NEWBORN(long N_NEWBORN) {
    this.N_NEWBORN = N_NEWBORN;
}

public long getN_IS_TEST() {
    return N_IS_TEST;
}

public void setN_IS_TEST(long N_IS_TEST) {
    this.N_IS_TEST = N_IS_TEST;
}

public long getN_IS_MARKED() {
    return N_IS_MARKED;
}
```

```
public void setN_IS_MARKED(long N_IS_MARKED) {
    this.N_IS_MARKED = N_IS_MARKED;
}

public long getN_EMPTY() {
    return N_EMPTY;
}

public void setN_EMPTY(long N_EMPTY) {
    this.N_EMPTY = N_EMPTY;
}

public long getN_W1ST_NOACCENTS() {
    return N_W1ST_NOACCENTS;
}

public void setN_W1ST_NOACCENTS(long N_W1ST_NOACCENTS) {
    this.N_W1ST_NOACCENTS = N_W1ST_NOACCENTS;
}

public long getN_UNKW_FIRST() {
    return N_UNKW_FIRST;
}

public void setN_UNKW_FIRST(long N_UNKW_FIRST) {
    this.N_UNKW_FIRST = N_UNKW_FIRST;
}

public long getN_1ST_LESS_C() {
    return N_1ST_LESS_C;
}

public void setN_1ST_LESS_C(long N_1ST_LESS_C) {
    this.N_1ST_LESS_C = N_1ST_LESS_C;
}

public long getN_LAST_LESS_C() {
    return N_LAST_LESS_C;
}

public void setN_LAST_LESS_C(long N_LAST_LESS_C) {
    this.N_LAST_LESS_C = N_LAST_LESS_C;
}

public long getN_1CONS() {
    return N_1CONS;
}

public void setN_1CONS(long N_1CONS) {
    this.N_1CONS = N_1CONS;
}
```

```
}

public long getN_INVAL_NAME() {
    return N_INVAL_NAME;
}

public void setN_INVAL_NAME(long N_INVAL_NAME) {
    this.N_INVAL_NAME = N_INVAL_NAME;
}

public long getN_LESS_W() {
    return N_LESS_W;
}

public void setN_LESS_W(long N_LESS_W) {
    this.N_LESS_W = N_LESS_W;
}

public long getN_BEG_HIPHEN() {
    return N_BEG_HIPHEN;
}

public void setN_BEG_HIPHEN(long N_BEG_HIPHEN) {
    this.N_BEG_HIPHEN = N_BEG_HIPHEN;
}

public long getN_END_HIPHEN() {
    return N_END_HIPHEN;
}

public void setN_END_HIPHEN(long N_END_HIPHEN) {
    this.N_END_HIPHEN = N_END_HIPHEN;
}

public long getN_IBEF_HIPHEN() {
    return N_IBEF_HIPHEN;
}

public void setN_IBEF_HIPHEN(long N_IBEF_HIPHEN) {
    this.N_IBEF_HIPHEN = N_IBEF_HIPHEN;
}

public long getN_IAFT_HIPHEN() {
    return N_IAFT_HIPHEN;
}

public void setN_IAFT_HIPHEN(long N_IAFT_HIPHEN) {
    this.N_IAFT_HIPHEN = N_IAFT_HIPHEN;
}

public long getN_BEG_SPC() {
```

```
        return N_BEG_SPC;
    }

    public void setN_BEG_SPC(long N_BEG_SPC) {
        this.N_BEG_SPC = N_BEG_SPC;
    }

    public long getN_END_SPC() {
        return N_END_SPC;
    }

    public void setN_END_SPC(long N_END_SPC) {
        this.N_END_SPC = N_END_SPC;
    }

    public long getN_INVALID_CHS() {
        return N_INVALID_CHS;
    }

    public void setN_INVALID_CHS(long N_INVALID_CHS) {
        this.N_INVALID_CHS = N_INVALID_CHS;
    }

    public long getN_SPC_TOGHT() {
        return N_SPC_TOGHT;
    }

    public void setN_SPC_TOGHT(long N_SPC_TOGHT) {
        this.N_SPC_TOGHT = N_SPC_TOGHT;
    }
}
```

8.2.5. Output.java

```

package NameQuality;

import java.io.File;
import java.io.FileWriter;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.io.*;
import java.text.SimpleDateFormat;
import java.util.Date;

public class Output {
    private static Output instance;

    FileWriter writeFile;
    FileWriter f_02kn_fn_no_accents;
    FileWriter f_03unk_fn;
    FileWriter f_04fn_lc;
    FileWriter f_05ln_lc;
    FileWriter f_06_1cons;
    FileWriter f_07inv_n;
    FileWriter f_08lwords;
    FileWriter f_09b_hif;
    FileWriter f_10e_hif;
    FileWriter f_11inv_bhif;
    FileWriter f_12inv_ahif;
    FileWriter f_13b_spc;
    FileWriter f_14e_spc;
    FileWriter f_15inv_c;
    FileWriter f_16spc_tg;
    FileWriter f_17no_errors;

    public Output (boolean ALL_OUTPUT, boolean OUTPUT_REPORT) throws Exception
    {
        BaseFiles bf = BaseFiles.getInstance();

        /* *****
        String                                outputPath                                =
"/HLTSYS/_nq_work/estadistica/bd1/output_results/"+(bf.VT_MIN_WORDS-
1)+"words/it2_a"+bf.VT_ACCENTS+"/";
        //                                outputPath    =    "/teste/ToDo/bd0_"+(VT_MIN_WORDS-
1)+"w_i1_a"+VT_ACCENTS+"/";
        //    outputPath = "/teste/ToDo/";
        ***** */

        //    Path filePath = Paths.get(System.getProperty("user.dir"));
        // filePath = Paths.get("/Program Files/Mirth Connect/");

```



```

//                                     filePath =
filePath.resolve("./projects/namequality/output_results/" + (bf.VT_MIN_WORDS-
1) + "words/accents" + bf.VT_ACCENTS + "/");
//                                     Path filePath = Paths.get("/Program Files/Mirth
Connect/projects/namequality/output_results/" + (bf.VT_MIN_WORDS-
1) + "words/accents" + bf.VT_ACCENTS + "/");

//String strPath = "C:/Program Files/Mirth
Connect/projects/namequality/output_results";
String outputPath = "/namequality/output_results/";
File dir = new File(outputPath);
boolean isDirCreated = dir.mkdirs();
if(isDirCreated)
    System.out.println("Directory created: " + outputPath);
// else
//     System.out.println("Failed to create directory.");

File writeFx;
if (OUTPUT_REPORT || ALL_OUTPUT) {
    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-
dd_HH:mm:ss");
    Date date = new Date();
    writeFx = new File(outputPath + formatter.format(date) + "_report.txt");
    this.writeFile = new FileWriter(writeFx);
}

if (ALL_OUTPUT) {
    writeFx = new File(outputPath + "f_02kn_fn_no_accents.txt");
    this.f_02kn_fn_no_accents = new FileWriter(writeFx);
    writeFx = new File(outputPath + "f_03unk_fn.txt");
    this.f_03unk_fn = new FileWriter(writeFx);
    writeFx = new File(outputPath + "f_04fn_lc.txt");
    this.f_04fn_lc = new FileWriter(writeFx);
    writeFx = new File(outputPath + "f_05ln_lc.txt");
    this.f_05ln_lc = new FileWriter(writeFx);
    writeFx = new File(outputPath + "f_06_1cons.txt");
    this.f_06_1cons = new FileWriter(writeFx);
    writeFx = new File(outputPath + "f_07inv_n.txt");
    this.f_07inv_n = new FileWriter(writeFx);
    writeFx = new File(outputPath + "f_08lwords.txt");
    this.f_08lwords = new FileWriter(writeFx);
    writeFx = new File(outputPath + "f_09b_hif.txt");
    this.f_09b_hif = new FileWriter(writeFx);
    writeFx = new File(outputPath + "f_10e_hif.txt");
    this.f_10e_hif = new FileWriter(writeFx);
    writeFx = new File(outputPath + "f_11inv_bhif.txt");
    this.f_11inv_bhif = new FileWriter(writeFx);
    writeFx = new File(outputPath + "f_12inv_ahif.txt");
    this.f_12inv_ahif = new FileWriter(writeFx);
    writeFx = new File(outputPath + "f_13b_spc.txt");
}

```

```

        this.f_13b_spc = new FileWriter(writeFx);
        writeFx = new File(outputPath + "f_14e_spc.txt");
        this.f_14e_spc = new FileWriter(writeFx);
        writeFx = new File(outputPath + "f_15inv_c.txt");
        this.f_15inv_c = new FileWriter(writeFx);
        writeFx = new File(outputPath + "f_16spc_tg.txt");
        this.f_16spc_tg = new FileWriter(writeFx);
        writeFx = new File(outputPath + "f_17no_errors.txt");
        this.f_17no_errors = new FileWriter(writeFx);
    }
}

static {
    try {
        BaseFiles bf = BaseFiles.getInstance();
        instance = new Output(bf.ALL_OUTPUT, bf.OUTPUT_REPORT);
    } catch (Exception e) { }
}

/** Static 'instance' method */
public static Output getInstance() {
    return instance;
}

public void closeFiles() throws Exception {
    BaseFiles bf = BaseFiles.getInstance();

    if (bf.OUTPUT_REPORT || bf.ALL_OUTPUT) {
        this.writeFile.close();
        /*****
        String outputPath = "/namequality/output_results/";
        File writeFx = new File(outputPath + "namequality_report.txt");
        writeFx.renameTo(new File(outputPath + "t.txt"));
        *****/
    }

    if (bf.ALL_OUTPUT) {
        this.f_02kn_fn_no_accents.close();
        this.f_03unk_fn.close();
        this.f_04fn_lc.close();
        this.f_05ln_lc.close();
        this.f_06_1cons.close();
        this.f_07inv_n.close();
        this.f_08lwords.close();
        this.f_09b_hif.close();
        this.f_10e_hif.close();
        this.f_11inv_bhif.close();
        this.f_12inv_ahif.close();
        this.f_13b_spc.close();
        this.f_14e_spc.close();
        this.f_15inv_c.close();
    }
}

```

```
        this.f_16spc_tg.close();
        this.f_17no_errors.close();
    }
}
```

